ARL-TR-8194 ● OCT 2017

US Army Research Laboratory

Radio Frequency Ranging for Swarm Relative Localization

by Jacob R Lockspeiser, Michael L Don, and Moshe Hamaoui

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-8194 ● OCT 2017

US Army Research Laboratory

Radio Frequency Ranging for Swarm Relative Localization

by Jacob R Lockspeiser
Drexel University, Philadelphia, PA

Michael L Don and Moshe Hamaoui
Weapons and Materials Research Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
October 2017	Technical Report	1–30 June 2017

4. TITLE AND SUBTITLE
Radio Frequency Ranging for Swarm Relative Localization

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Jacob R Lockspeiser, Michael L Don, and Moshe Hamaoui

5d. PROJECT NUMBER
23172.8.3901

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
US Army Research Laboratory
ATTN: RDRL-WML-F
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-8194

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Swarms of agents exhibit advantages over a comparable group of solitary agents. One advantage is the ability for swarm agents to localize relative to the group using spatial relationships between many agents to achieve accurate relative position information. This is particularly important in GPS-denied environments where there are limited positioning options. Many applications exist for relative positioning, such as collision avoidance, formation flying, and patterned weapon delivery. Although there are many technologies that can be employed for relative localization, this report focuses on Atmel and Nanotron narrowband RF ranging products. Atmel transceivers are briefly evaluated but do not meet the US Army Research Laboratory's (ARL's) sampling rate and range requirements. Nanotron transceivers are more thoroughly evaluated and are shown to be promising products for ARL's swarm localization needs. The results of successful swarm localization experiments with 6 agents are presented.

15. SUBJECT TERMS
RF ranging, swarm localization, swarm networking, Nanotron, Atmel

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Michael L Don
Unclassified	Unclassified	Unclassified	UU	64	19b. TELEPHONE NUMBER (Include area code) 410-306-0775

Contents

List of Figures

List of Tables

Acknowledgments

The authors would like to acknowledge Daniel Everson of the US Army Research Laboratory for his help in recording the survey data for the outdoor dual Nanotron range testing as well as other help with testing setup. Barry Kline of SURVICE Engineering Company also assisted with the swarm localization test setup. John Hallameyer of Bowhead provided support with various aspects of the embedded systems.

1. Introduction

Swarms of agents exhibit advantages over a comparable group of solitary agents. One advantage is the ability for swarm agents to localize relative to the group, using spatial relationships between many agents to achieve accurate relative position information.[1] This is particularly important in GPS-denied environments where there are limited positioning options.[2] Many applications exist for relative positioning such as collision avoidance,[3] formation flying,[4] and patterned weapon delivery.[5] In addition, relative localization can be transformed into absolute localization even if the absolute positions of only a few agents are known. There are many techniques that can be employed for relative localization including RF, ultrasound, and optical technologies.[6] This report focuses on evaluating RF 2-way ranging (TWR) products for swarm localization.

Ultrasound technologies have been demonstrated to be very accurate, but their typical maximum range of only a few meters is unsuitable for many swarm applications.[7] A variety of optical systems also exist for range measurements. Laser-based systems are very accurate but typically have a small field of view (FOV). More recently, time-of-flight cameras have been developed that offer a wider FOV but suffer from smaller measurement ranges.[8] Stereo cameras can be used for ranging,[9] as well as single cameras through measuring the size of known markers,[10] but the hardware and image processing requirements make integration into small embedded systems problematic. In contrast to other technologies, RF ranging usually has a large FOV and long range. Aside from RF TWR, localization can also be accomplished using RF angle of arrival (AOA), time of arrival (TOA), and time difference of arrival (TDOA). AOA systems require calibrated antenna arrays that limit the availability of suitable commercial solutions.[11] TOA and TDOA systems have also been shown to provide reliable ranging results, but they require specialized infrastructure to create the necessary timing synchronization between agents.[12] Distance can also be estimated using a received signal strength indicator (RSSI), but this method typically has low accuracy.[13]

Compared with other technologies, RF TWR products have many benefits, including the following[14]:

- Low-cost commercial products
- Low power
- Small size
- Accuracy on par with GPS

- Wide FOV
- Long range
- Supports wireless communications in addition to ranging

A few disadvantages include susceptibility to jamming and interference plus possible antenna integration problems on small custom platforms such as munitions. Considering that a swarm will most likely have wireless networking capabilities, there is no reason not to use the RF signals for ranging as well as communications. RF TWR products generally fall into 2 categories, ultra-wideband (UWB) and narrowband (NB). UWB products are more accurate, but due to transmit (TX) power limitations, have a shorter range.[15] NB products are less accurate but have longer ranges that can be extended even further through external amplification. In addition, frequency hopping to prevent jamming and interference is theoretically easier with NB ranging because of the greater number of available frequency slots. Due to the benefits of NB ranging, 2 major NB ranging commercial products were chosen for evaluation, the Atmel REB233SMAD[16] and the Nanotron Swarm BEE LE (low energy).[17]

The performance goal for evaluating these products is to provide localization on par with current GPS capabilities, both in accuracy and update rate. The short-term maximum range objective is 100 m, while the long-term goal is ranging out to 1 km. In ranging applications, agent position accuracy depends on swarm geometry and is therefore difficult to characterize, but in general individual ranging errors are on the order of a few meters.[18] Update rates can also be difficult to characterize because the ranging channel may have to be shared with communications, but requirements can be estimated. Assuming that only one agent can perform a ranging operation at a time, and that the distances between all of the agents are required to perform localization, the ranging measurement period, P_a, to support a total swarm localization update rate of R_s for a swarm of size N is

$$P_a = \frac{(N-2)!2}{N!R_s}. \quad (1)$$

The top plot of Fig. 1 illustrates how quickly the number of range measures grows with swarm size. The bottom plot shows an example total swarm localization update rate calculation for P_a= 10 ms. The update rate quickly drops as the swarm size increases, showing the importance of RF ranging measurement speed. Initial research focused on a modest swarm size of 6 agents with an update rate goal of 5 Hz. This gives a maximum measurement period of $P_a = 13.3$ ms.

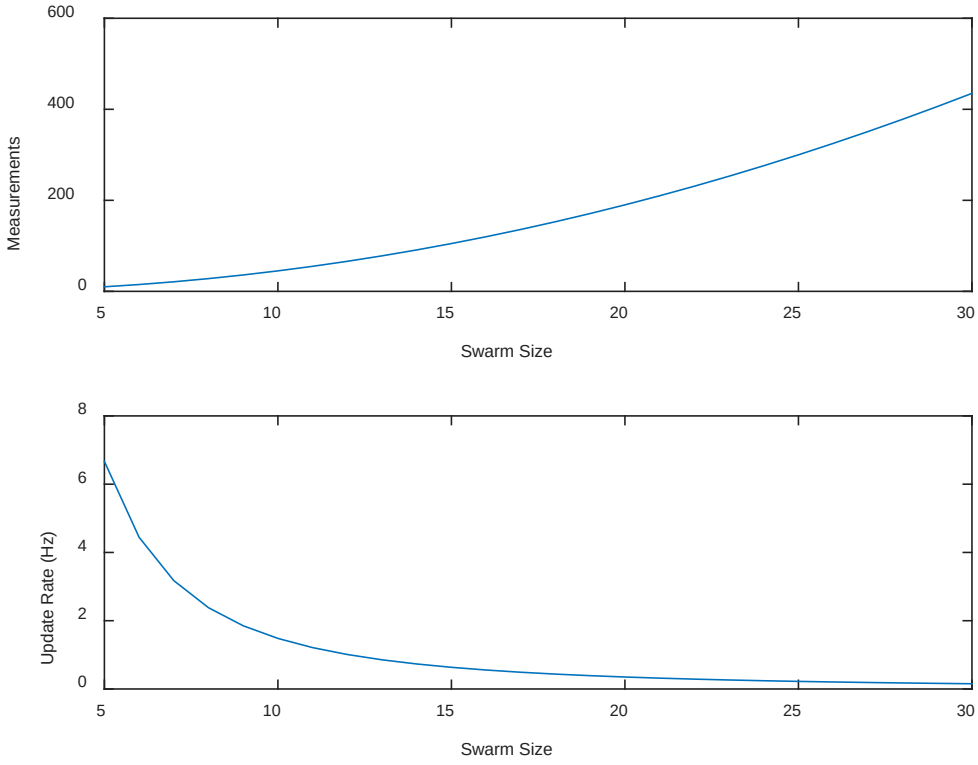

Fig. 1 Total number of swarm distance measurements (top) and the total swarm localization update rate (bottom) for a give swarm size

The Atmel product is introduced first, briefly evaluated, and then rejected for swarm localization. Then the Nanotron product is more thoroughly evaluated. A networking scheme for swarm ranging is presented. To test the feasibility of this scheme, an experiment setup using 2 Nanotron kits is developed. Test data in both a controlled laboratory environment and an outdoor setting are presented and analyzed. Lastly, an experiment setup is designed for the swarm localization of 6 agents. Localization techniques are discussed, and the experiment data are presented and analyzed.

2. Atmel Evaluation

Atmel manufactures the AT86RF233 radio transceiver, which uses a phase difference measurement unit (PMU) for RF ranging. It operates in the 2.4-GHz industrial, scientific and medical (ISM) radio band and conforms to the Institute of Electrical and Electronics Engineers 802.15.4-2006/2011 standard.[16] Some of the characteristics of the AT86RF233 transceiver include the following:

- 105-dB link budget

- RSSI measurement, energy detection, and link quality indication
- Advanced encryption standard 128-bit hardware accelerator
- Antenna diversity and TX indication
- Supported data rates: 250, 500, 1000, and 2000 kb/s
- Time and phase measurement support
- 32-pin low-profile package: 5 × 5 × 0.9 mm^3

Atmel provides the REB233SMAD development kit for the AT86RF233 transceiver to demonstrate the functionality of the PMU and evaluate the radio transceiver performance (Fig. 2). The kit also contains an ATxmega256A3 microcontroller, battery power, and dual antennas. Software support includes a ranging toolbox library and an evaluation application. Custom programs can be developed using Atmel Studios integrated development environment (IDE).

Fig. 2 REB233SMAD development kit

The evaluation application uses 3 REB233SMAD kits designated as a Coordinator, Initiator, and Reflector. The Coordinator controls the other 2 kits and is connected to a PC. The Initiator and the Reflector operate in stand-alone mode and perform the ranging operations. The example application provides a number of programmable settings including the following[19]:

- Start frequency, step frequency, and stop frequency selection

- TX power
- Antenna diversity control
- Filter settings
- Addressing settings

Ranging operations are performed at each frequency step and then averaged, with the total number of steps denoted as N_s. Since multipath is frequency dependent, frequency diversity aids in multipath mitigation. A single ranging operation begins with the Initiator and is reflected back by the Reflector, giving a distance measurement of

$$d = \frac{c(t_{rt} - t_d)}{2}, \quad (2)$$

where c is the speed of light, t_{rt} is the total round trip ranging time, and t_d is a fixed system delay. Increasing N_s will increase the accuracy of the ranges but will also increase the ranging measurement time.

2.1 Atmel Sample Rate Testing

Two problems were quickly identified with the Atmel devices. First, the average ranging sampling period of 172 ms was much slower than the goal of 13.3 ms. Second, we experienced difficulty in obtaining long-range outdoor measurements. Due to these problems, only a limited amount of testing was performed. The first test used 3 different values of N_s in an attempt to characterize slow, medium, and fast ranging sample rates. Table 1 shows the frequency parameters used for this test. Two Atmel kits were placed 460 cm apart, the parameters were programmed using the example application, and data were collected using a custom LabVIEW program. This program displayed the range measurements, plotted the ranges in real time, and saved the range data for further postprocessing.

Table 1 Sampling rate test parameters

Sampling rate	Start frequency (MHz)	Step frequency (MHz)	Stop frequency (MHz)	N frequency samples
Fast	2324	2	2328	2
Medium	2403	2	2443	20
Slow	2324	0.5	2527	406

Figure 3 shows the fast, medium, and slow sampling rate results along with the actual measured distance. Summary statistics are listed in Table 2. The fast sampling rate, which only used 2 frequency steps, performed very poorly, with an

average range error of 360%. Figure 4 is a close-up of Fig. 3, showing the medium and slow sampling rates in more detail. The slow sampling rate results are very accurate, averaging less than a 1% error with a standard deviation of approximately 17.2 cm over 406 samples. The medium sampling rate was set to the example program's default frequency parameters of 20 samples in the Wi-Fi frequency range between 2.4 and 2.484 GHz. This resulted in a 20.5% error with the standard deviation doubling from the slow case to 34.2 cm. As expected, these tests showed the error decreasing as N_s increased. However, the sampling rate did not scale linearly with N_s. The maximum sampling rate we achieved in the fast sampling case was about 6.2 Hz, only slightly higher than the medium rate of 5.8 Hz. On the other hand, the slow sampling rate was measured at 3.2 Hz, faster than expected considering $N_s = 406$. These discrepancies may be due to a limitation in the example application, and not in the hardware itself, but further analysis was not performed at this time.

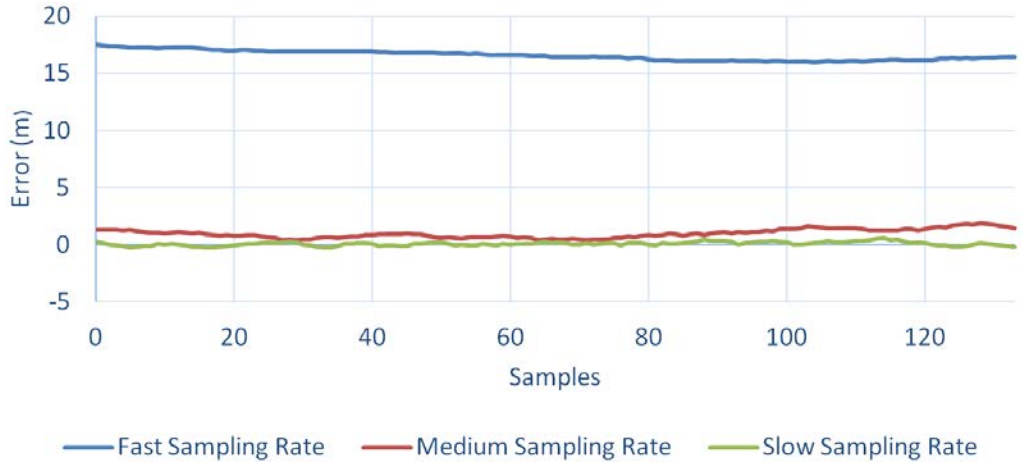

Fig. 3 Atmel sampling rate tests results

Table 2 Atmel sampling rate test summary

Sampling rate	Rate (Hz)	Average (cm)	Error (%)	Std. deviation (cm)
Fast	6.2	2118	360.3	42
Medium	5.8	554	20.5	34
Slow	3.2	464	0.9	17

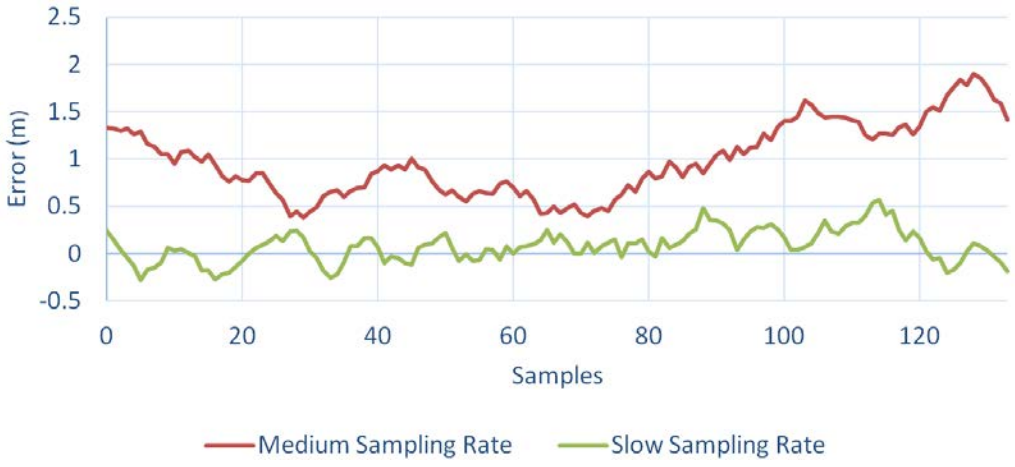

Fig. 4 Zoomed medium and slow sampling rate test results

2.2 Atmel Frequency Band Testing

Due to the possibility of Wi-Fi interference affecting the results of Atmel ranging, it was beneficial to characterize the ranging performance in frequency bands outside of the Wi-Fi range. Table 3 shows the parameters used in this test. The default frequency range is the Wi-Fi band from 2.4 to 2.484 GHz. Two other bands were chosen for testing, one below and one above the Wi-Fi band. N_s was kept nearly constant throughout the tests, but Atmel frequency options caused a slight deviation for the frequency band above Wi-Fi. The number of samples differing by one, however, should not change results by a noticeable factor. The medium sampling rate case uses the default parameters with the frequency range directly within the Wi-Fi band. As seen in Fig. 5 and Table 4, the range below Wi-Fi has a lower average error of 8.16%, but the range above Wi-Fi has a lower deviation. The default case performs in between the results of the above and below Wi-Fi cases. In general, the results of these 3 cases were comparable, allowing the use of other frequency bands to avoid Wi-Fi interference.

Table 3 Wi-Fi frequency test parameters

Parameter	Start frequency (MHz)	Step frequency (MHz)	Stop frequency (MHz)	Number of frequency samples
Default	2403	2	2443	20
Below Wi-Fi	2360	2	2400	20
Above Wi-Fi	2486	2	2524	19

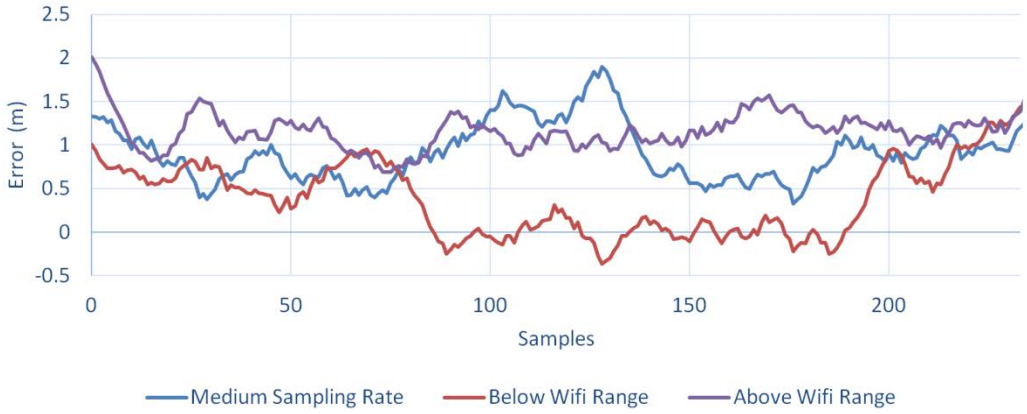

Fig. 5 Wi-Fi frequency test results

Table 4 Wi-Fi frequency test summary

Parameter	Average (cm)	% error	Std. deviation (cm)
Default	554	20.5	34
Below Wi-Fi	498	8.2	42
Above Wi-Fi	583	26.7	25

2.3 Atmel Distance Testing

The next experiment used the default frequency parameters and tested the Atmel kits at various distances. The distances were measured with a laser range finder and set at 8, 17.2, and 24.5 m. Laser range finders themselves have an accuracy of 0.5 m,[20] which may have contributed to the calculated Atmel error. Even with the TX power set to the maximum level, we experienced difficulties ranging at distances greater than 25 m. This problem was not thoroughly investigated because the Nanotron devices are clearly more suitable for ARL's swarm localization requirements. Figure 6 shows the data for all 3 tests with the percent error and deviation shown in Table 5. The accuracy of these results were promising, with low deviations and relatively small errors.

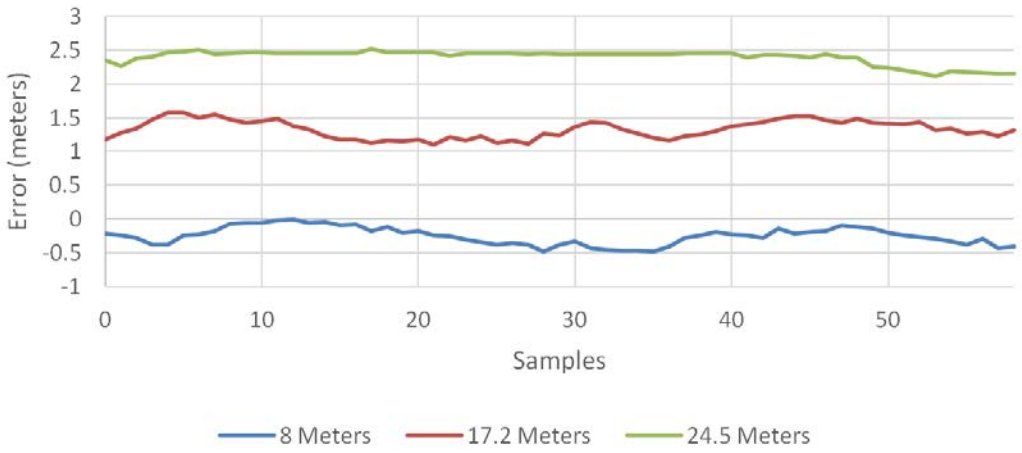

Fig. 6 Distance testing results

Table 5 Distance test summary

Specification	8 m	17.2 m	24.5 m
Average (m)	7.74	18.53	26.90
% error	3.30	7.16	8.91
St. dev. (m)	0.13	0.13	0.10

Although the accuracy of the Atmel kits was acceptable, the sampling period of about 172 ms was much slower than the 13.3-ms goal mentioned in the Introduction. This problem, combined with the measurement difficulty experienced for long-range outdoor measurements, resulted in ARL rejecting Atmel ranging products for swarm localization research.

3. Nanotron Evaluation

To improve ranging performance over the Atmel kits, Nanotron Swarm Bee LE kits, shown in Fig. 7, were acquired. The Nanotron modules contain a microcontroller for control and interfacing aside from the Nanotron radio transceiver itself. Ranging uses a chirp spread spectrum (CSS) NB signal in the 2.4- to 2.4835-GHz Wi-Fi range with a selectable data rate of 1 or 0.25 Mbs. The advantages of CSS ranging include high ranging resolution and substantial resistance to multipath interference. The Nanotron's maximum transmission power is 16 dBm with a link budget of 105 or 111 dB for the 1- and 0.25-Mbs data rate modes, respectively. The configuration and communication software included with the kits facilitates system integration.[17]

Fig. 7 Nanotron Swarm BEE LE kit

Evaluation of the Nanotron kits quickly showed that they possessed the capabilities that the Atmel kits lacked. Whereas Atmel had a slow sampling rate and limited range, the Nanotron kits worked easily at longer ranges at high sampling rates, making them promising candidates for swarm localization.

3.1 Nanotron Networking

Evaluating the ranging operations alone is not sufficient to determine the Nanotron kits' suitability for swarm localization. To implement an RF ranging localization scheme, the ranging information must be communicated to other swarm agents. The Nanotron communication medium, however, uses the same medium as the ranging operations. Thus, a medium access control (MAC) protocol must be used to avoid collisions between ranging and communication operations.

One simple MAC protocol is time-division multiple access (TDMA), in which each ranging and communication operation is assigned a separate time slot to prevent transmission collisions. Example TDMA state diagrams for a swarm of 3 agents are illustrated in Fig. 8. Rij indicates a ranging operation is performed from agent i (Ai) to Aj. Bij indicates a range broadcast where Ai broadcasts the results of Rij to the entire swarm. For 3 agents, there are a total of 3 range measurements: R12, R13, and R23 with their corresponding broadcasts B12, B13, and B23. The state diagram of A1 is on the left, A2 is on the right, and A3 is omitted since it does not initiate any ranging operations. The actions performed in each state are written inside the state bubbles, and state transition logic is written next to the state transition arrows. Dashed arrows are used to illustrate a causal relationship. States that perform a ranging operation are shaded light orange while the other states are

white. This makes the TDMA scheme clear: only one agent is allowed to perform a ranging operation at a time. In this example, each combined ranging and broadcast operation takes a little less than 15 ms.

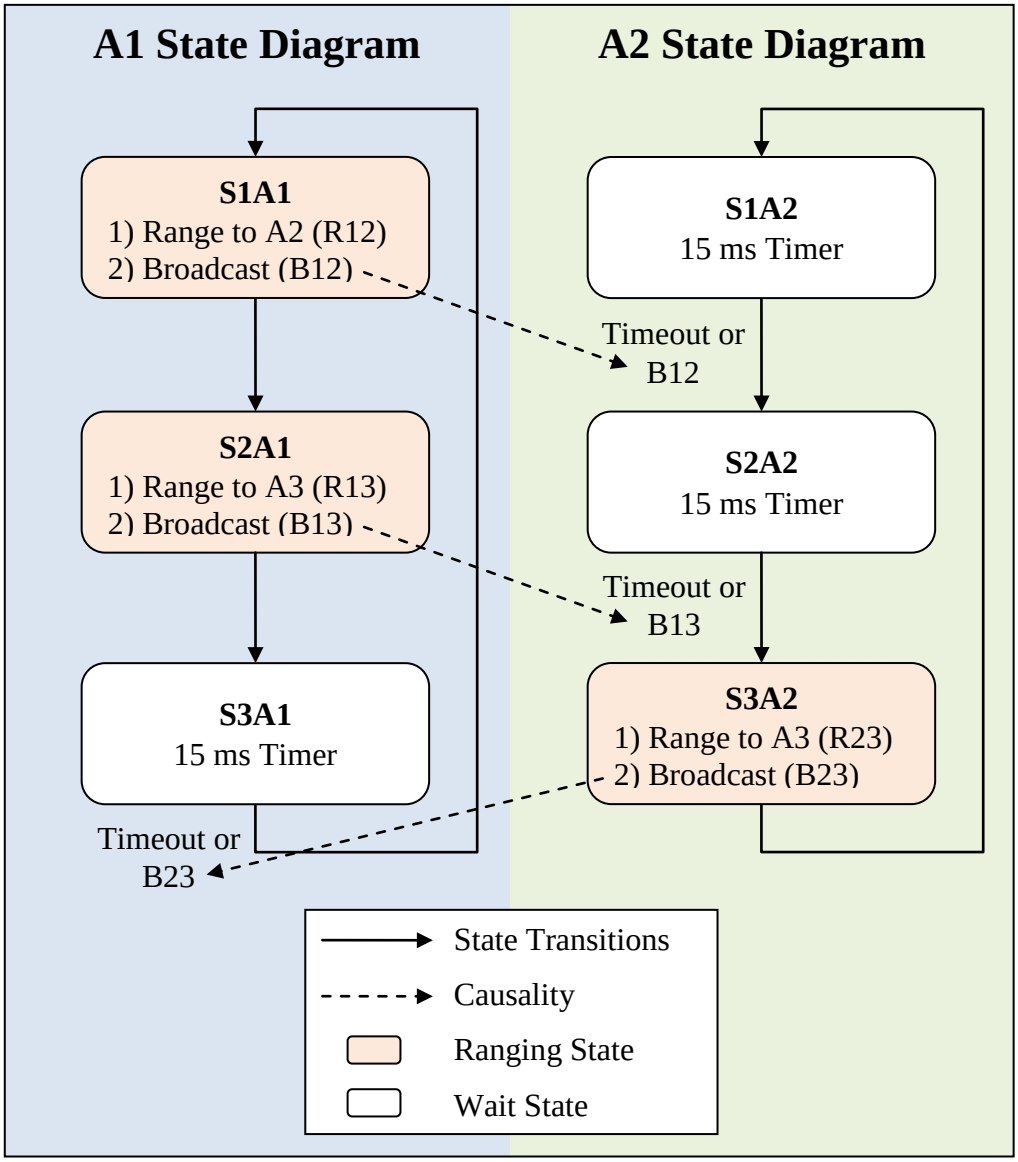

Fig. 8 TDMA state diagram of agents 1 and 2 for a swarm of 3 agents

Starting in the upper left with the first state of A1 (S1A1), A1 performs R12 and B12. During this time A2 is also in its first state (S1A2) waiting to move to S2A2. This occurs when A2 receives B12 from A1 (indicated by a dashed arrow) or A2's 15-ms timer expires. The timer is required to ensure that A2 will eventually move to S2A2 even if it misses the B12 transmission. The timer alone cannot be used for state transitions because the clock drift of each agent would cause the state

machines to become unsynchronized over time. Therefore, the broadcasts must also be used to trigger the state transitions to ensure that the A1 and A2 state machines remain synchronized. Once A1 finishes R12 and B12, it moves to S2A1 and performs R13 and B13. As before, A2 waits in S2A2 for either B13 or a timeout to move to S3A2. Once A1 moves to S3A1 and A2 moves to S3A2, they switch roles. A2 now performs R23 and B23, and A1 waits for either B23 or a timeout to move to S1A1. In summary, each agent takes turns ranging to avoid collisions. State transitions are triggered by broadcasts to maintain swarm synchronization while relying on timers in case the broadcasts are missed.

This scheme can be generalized for any number of agents. For example, the states for a swarm of 4 agents with a total of 6 ranging operations are listed in a compact form in Table 6. Although this scheme is sufficient for a basic swarm localization scenario, further development in swarm networking could include additional factors such as agents leaving or entering the swarm, multihop communications, and the evaluation of carrier sense medium access protocols.

Table 6 TDMA scheduling example for 4 agents. Rij indicates a ranging operation from Ai to Aj with broadcast Bij.

State	A1	A2	A3
S1	R12, B12	15-ms timer	15-ms timer
S2	R13, B13	15-ms timer	15-ms timer
S3	R14, B14	15-ms timer	15-ms timer
S4	15-ms timer	R23, B23	15-ms timer
S5	15-ms timer	R24, B24	15-ms timer
S6	15-ms timer	15-ms timer	R34, B34

3.2 Nanotron Laboratory Testing

To verify the capabilities of the Nanotron kits, it was sufficient to only use 2 units in this TDMA scheme. An Arduino Mega 2650 controller[21] was used to control each Nanotron module through a 500-K-baud universal asynchronous receiver-transmitter (UART) with the program included in Appendix A. The program contains 2 main functions: setup and loop. The setup function runs once at startup and initializes all of the program's variables and Nanotron settings. It is assumed that the Nanotron IDs are preprogrammed and that one of the Nanotrons has an ID of 0, which will be referred to as A1. A1 is designated to begin the ranging operations. After setup completes, the loop function continually executes until power down. The loop begins with A1 ranging to the other Nanotron, A2, and then broadcasts the result. The Nanotron is configured to broadcast the range and unit IDs automatically after a ranging operation, but it is also possible to disable this

feature and perform a custom broadcast that includes additional data. Once A2 receives the broadcast, it ranges to A1 and broadcasts the result. A1 and A2 continually take turns ranging and broadcasting to one another. If one of the units waits for a broadcast for more than 25 ms, it times out and independently begins a new ranging operation. Figure 9 shows an oscilloscope screen capture of an example Nanotron ranging and broadcast operation. The light blue signal shows the initial Nanotron UART receiver (RX) receiving a command from the Arduino to perform a ranging operation. The actual ranging operation consists of several transmissions from A1 (pink) and responses from A2 (green). The final transmission from A1 is the broadcast message accompanied by the Nanotron UART TX of the ranging results. Not shown in Fig. 9, A2 will report the ranging results on its UART TX after receiving the broadcast. The next ranging operation by A2 can be observed at the next falling edge of A2 RF TX. The duration of one ranging and broadcast operation is about 14 ms, which is close to the 13.3-ms goal.

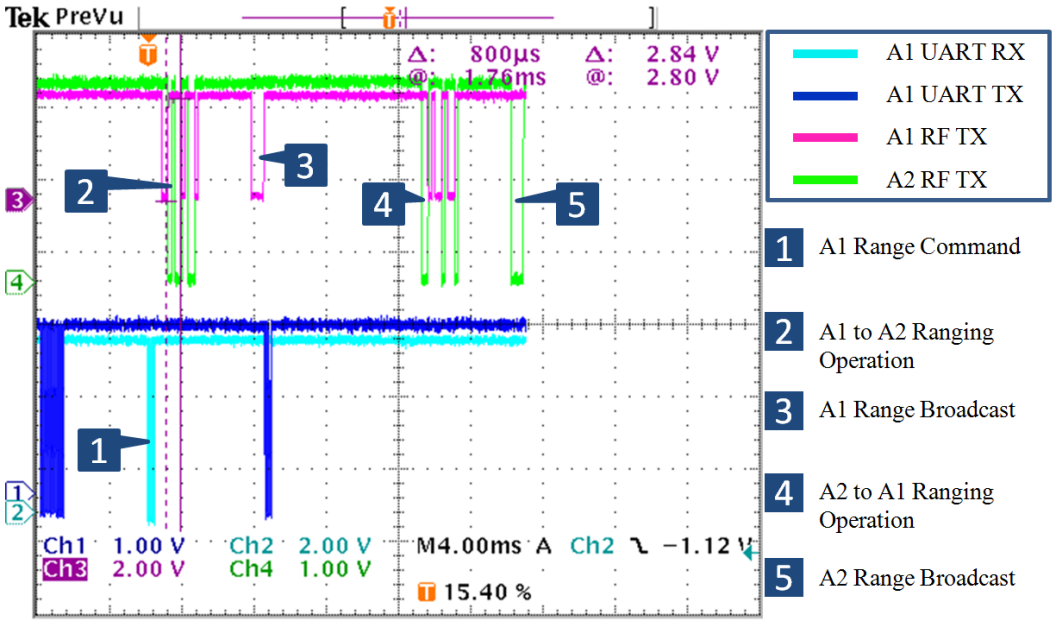

Fig. 9 Ranging and broadcast operation from agent A1 to A2

Once 2 Nanotron kits were configured to range to one another using the Arduino controllers, evaluation of the Nanotron modules began in a controlled laboratory environment. Figure 10 shows the testing setup. The 2 Nanotron kits with their Arduino controllers were placed in 2 separate anechoic chambers. Each was connected through a coaxial cable to a programmable attenuator. A PC was connected to unit A1 to record testing results.

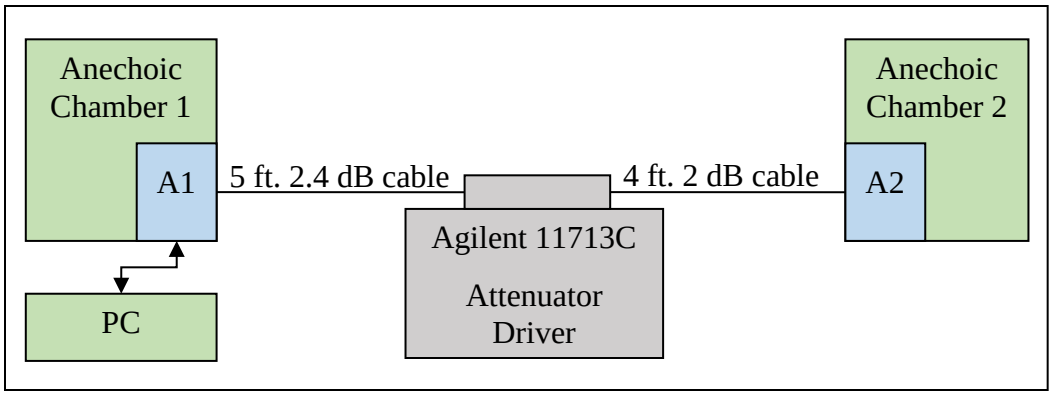

Fig. 10 Laboratory Nanotron evaluation setup

The first 3 experiments characterized ranging errors per second, cycle time, and accuracy versus attenuation using a TX power of –22 dBm. Figure 11 shows the errors per second versus attenuation, where error refers to an unsuccessful ranging operation. As is typical for wireless communications, performance falls off quickly after passing a given threshold, here at about 65 dB of attenuation. This attenuation, combined with the 4.4-dB cable loss shown in Fig. 10 and a 1-dB connector loss, brings the –22 dBm of transmit power down to –92.4 dBm, close to the specified receiver sensitivity in this mode of –89 dBm.

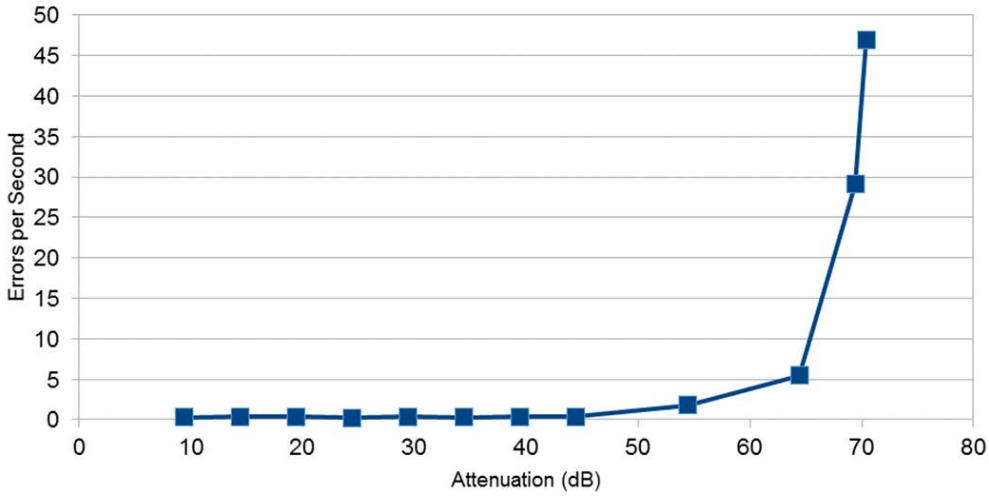

Fig. 11 Errors per second vs. attenuation

Figure 12 shows the average cycle time versus attenuation, where cycle time refers to the duration of a ranging and broadcast operation. When performing well, each ranging cycle by A1 and A2 are triggered by the previous broadcast. As the attenuation is increased and messages are dropped, more of the ranging operations are triggered by timeouts, increasing the average cycle time. At 70 dB of attenuation there was complete failure, indicated by a cycle time of 0.

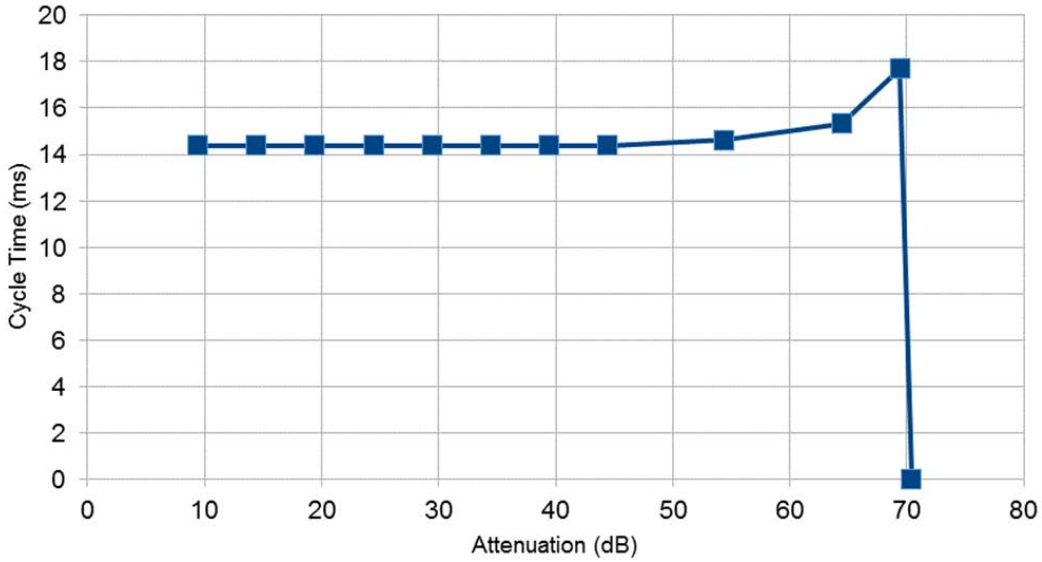

Fig. 12 Cycle time vs. attenuation

Figure 13 shows the percent range accuracy versus the attenuation. The usefulness of this test is limited because it was only performed through the 9 ft of cable, but more-extensive outdoor ranging at greater distances is presented in Section 3.3. The accuracy remains high compared with GPS standards, considering that a 70% accuracy at 9 ft is only a 2.7-ft error. Once again, at 70 dB of attenuation, all ranging attempts fail.

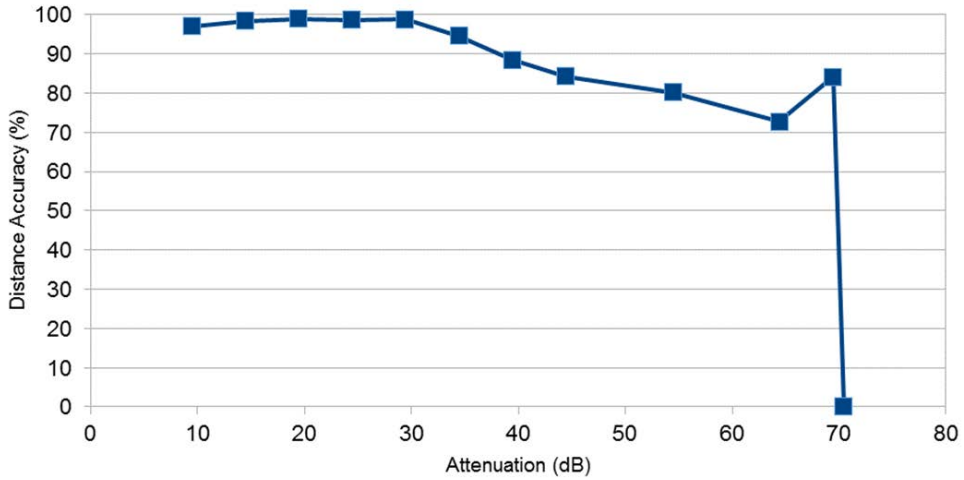

Fig. 13 Accuracy vs. attenuation

The next experiment tested the maximum attenuation before failure vs. TX power. Two modes affect the ranging performance. Switches Forward Error Correction (SFEC) is a Nanotron error correction command that adds error correction codes to the data frames, increasing the performance at the cost of the additional bits per frame. Set Data Mode (SDAM) is a Nanotron low data rate command, resulting in

a bit rate of 0.25 Mbs when enabled and 1 Mbs if disabled. The results of testing the 4 possible combinations of these 2 commands are shown in Fig. 14. By default, both SDAM and SFEC are disabled, giving the highest data rate but the lowest performance. At a TX power level of –22 dBm, the maximum attenuation before complete failure in the default mode is 70 dB, corresponding to the previous results in Figs. 11–13. In general, enabling SFEC provides about 1 dB of performance gain, while enabling SDAM provides a 5-dB gain.

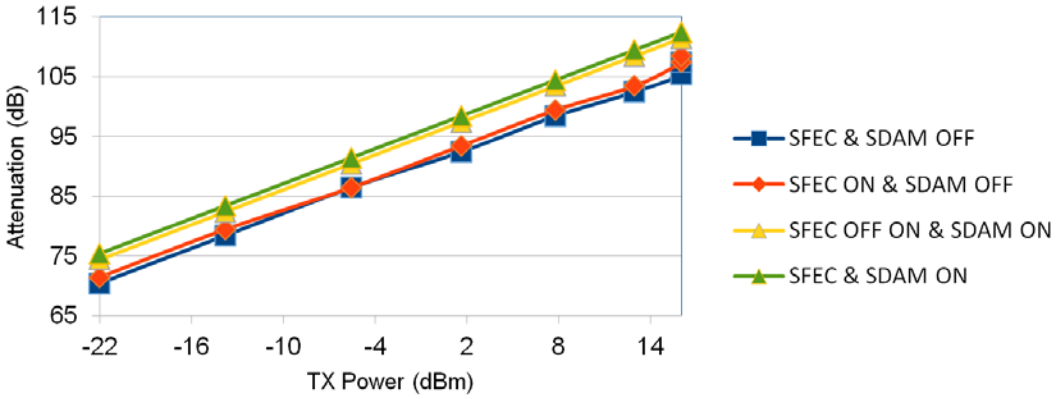

Fig. 14 Maximum attenuation vs. TX power

The attenuation in Fig. 14 can be converted to distance using equation for Free-Space Path Loss (FSPL)[22]:

$$FSPL = 20\log_{10}(d) + 20\log_{10}(f) - 20\log_{10}(4\pi/c). \quad (3)$$

Here $FSPL$ is in decibels, d is the distance, f is the radio frequency, and c is the speed of light. Solving for d gives

$$d = 10^{(FSPL/20 - \log_{10}(f) + \log_{10}(4\pi/c))}. \quad (4)$$

Using the attenuation from Fig. 14 plus additional cable and connector loss, the corresponding distances shown in Fig. 15 were calculated. These results are promising, predicting ranges out to 1 km in the default mode at only a TX power level of 10 dBm. These results should correspond well with high-altitude environments that can be considered free-space. In settings closer to the ground, a simplified 2-ray ground-reflection model should produce more-accurate results. This model calculates path loss (PL) as

$$PL = 40\log_{10}(d) - 10\log_{10}(Gh_t^2h_r^2), \quad (5)$$

where G is the antenna gain, h_t is the transmitter height, and h_r is the receiver height. Using $G = 1$ dB and $h_t = h_r = 1$ m, the distance is

$$d = 10^{PL/40}. \quad (6)$$

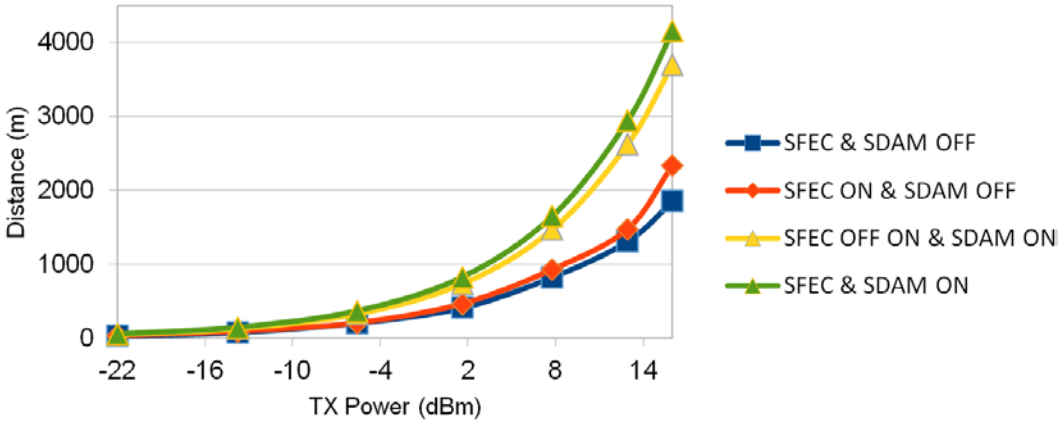

Fig. 15 Free-space distance vs. TX power

Figure 16 shows the distance calculated using the simplified 2-ray ground-reflection model. Using the default modes and maximum TX power, the predicted distance is about 500 m, corresponding well to the Nanotron-specified 500-m maximum range.

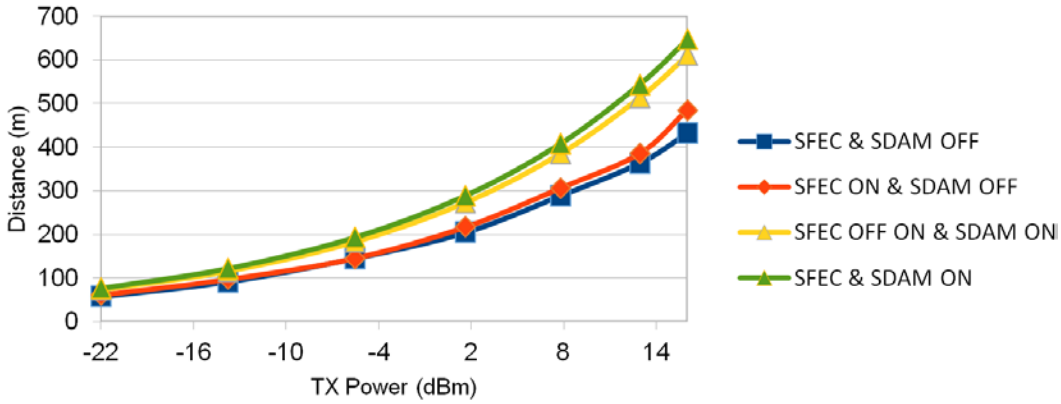

Fig. 16 Simplified 2-ray ground-reflection model distance vs. TX power

3.3 Dual Nanotron Outdoor Testing

After the laboratory evaluation, outdoor testing was conducted to characterize the Nanotron's performance at longer ranges. A stationary Nanotron kit and Arduino controller was connected to a PC running a LabVIEW program, which saved and timestamped the Nanotron's ranging data. The other Nanotron kit and Arduino controller were battery powered and mounted with a survey prism for easy transportation. The location of the stationary unit was surveyed using a Leica TS16 Total Station[23] and survey prism, and then the mobile unit was hand carried while recording the Nanotron and survey data. The Nanotron data were recorded at about a 10-Hz sampling rate, while the survey data were recorded at a 1-Hz rate. In the first test, the mobile unit was carried away from the stationary unit over fairly level

ground while maintaining line of sight and then returned back toward the stationary unit. All of these outdoor tests were conducted with SDAM and SFEC off and the TX power set to the maximum level. The top plot of Fig. 17 compares the Nanotron ranges with the survey ranges, with the calculated error displayed in the bottom plot. In general, the Nanotron ranges closely tracked the survey ranges with a few outliers evident.

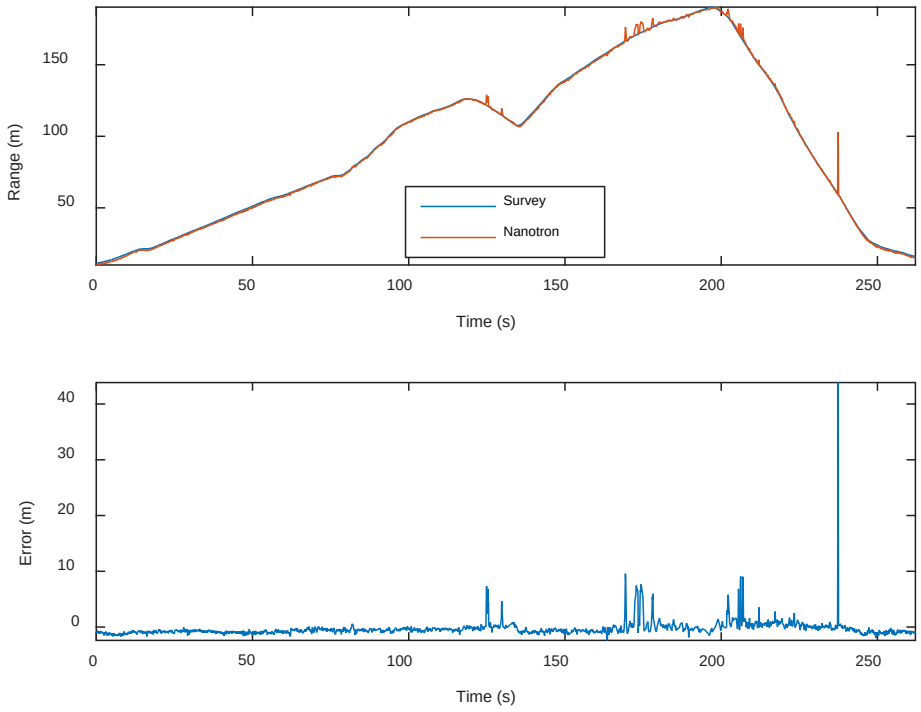

Fig. 17 Comparison of the Nanotron and survey ranges of the first outdoor test (top) with the calculated error (bottom)

More-precise statistics of this test are shown in the boxplot of the absolute value of the error in Fig 18. The range data were divided into 15 equal-sized range bins of 12 m each. The X-axis shows the midpoints of these range bins. For each bin, the red line indicates the mean error and the edges of the boxes indicate the 25th and 75th percentiles. The top whisker goes to $q3 + 1.5(q3 - q1)$, and the bottom whisker goes to $q1 - 1.5(q3 - q1)$, where $q1$ and $q3$ are the 25th and 75th percentiles, respectively. Any points outside the whiskers are considered outliers and are marked with a circle. All outliers past the dashed line at 4 m are plotted on the line to keep the plot at a viewable scale. The mean error is about 1 m with small variation except for the farthest measurements.

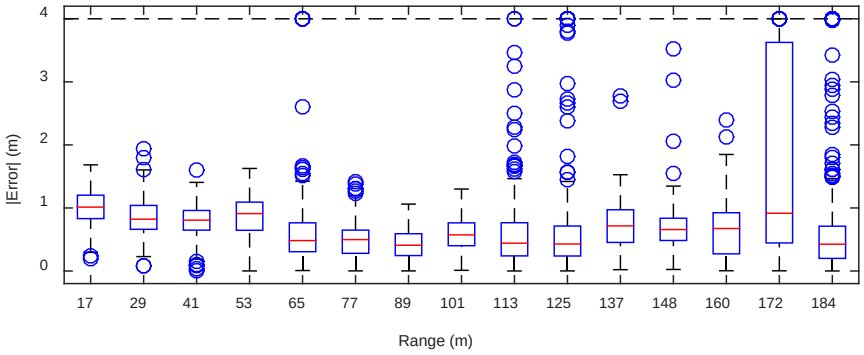

Fig. 18 Boxplot of the absolute value of the error of the first outdoor test

The top of Fig. 19 shows the dropped ranging operations with the RSSI displayed in the bottom plot. Both the RSSI reported by the Nanotron transceiver and predicted RSSI using Eq. 6 are displayed showing a strong correlation. The greatest dissimilarities are at short ranges, with the reported RSSI significantly lower than predicted. This occurs in all 3 outdoor tests but is of little concern since these RSSI levels are well within the Nanotron's sensitivity limits.

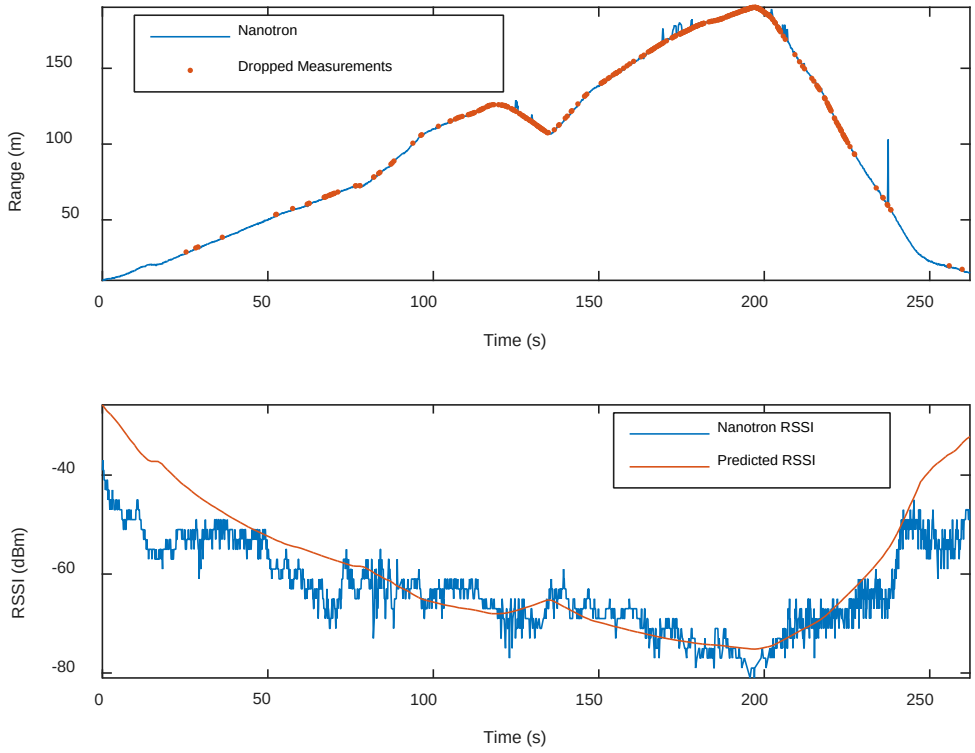

Fig. 19 Dropped Nanotron ranges (top) and RSSI (bottom) of the first outdoor test

The amount of dropped measurements can be interpreted easily from Fig. 20, which shows a histogram of the percentage of dropped measurements for various range bins. Past 100 m, the percentage of dropped measurements becomes significant, with up to half of the measurements dropped at the farthest ranges. These results are disappointing considering that the Nanotron kits specify a maximum range of 500 m, but specifications can be difficult to interpret. Is this maximum range with SDAM and SFEC on or off? Does it assume free-space or is it over ground? The only reliable way to answer these questions is through field tests.

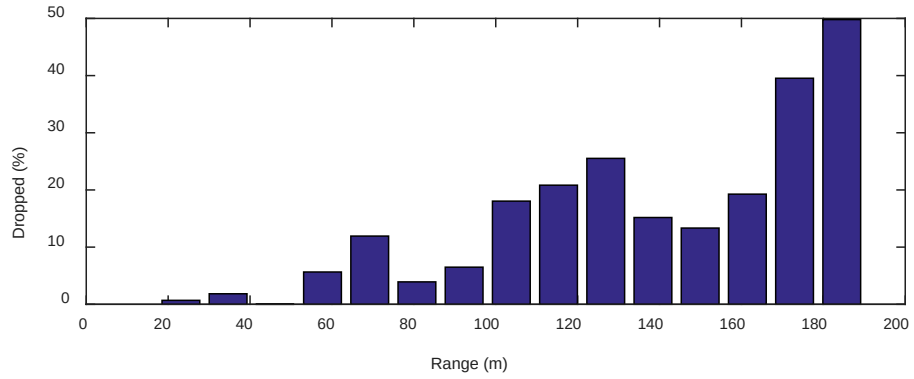

Fig. 20 Percentage dropped ranges vs. range of the first outdoor test

Another factor that may affect the Nanotron ranging performance is Wi-Fi interference. The Nanotron transceivers operate in the same frequency band as Wi-Fi, and even outdoors there is a significant amount of Wi-Fi interference. The Wi-Fi spectrum was measured using a RF signal analyzer both inside a building and outside where the Nanotron ranging tests were conducted. The top plot of Fig. 21 shows the indoor spectrum and the bottom plot shows the outdoor spectrum, whose highest amplitudes were only about 5 dB below those of the indoor spectrum. Additional testing is required to determine the degree to which the Wi-Fi signals interfere with the Nanotron ranging. In any case, ranging does work well out to the specified short term goal of 100 m. For greater distances, the signal may have to be amplified to increase performance.

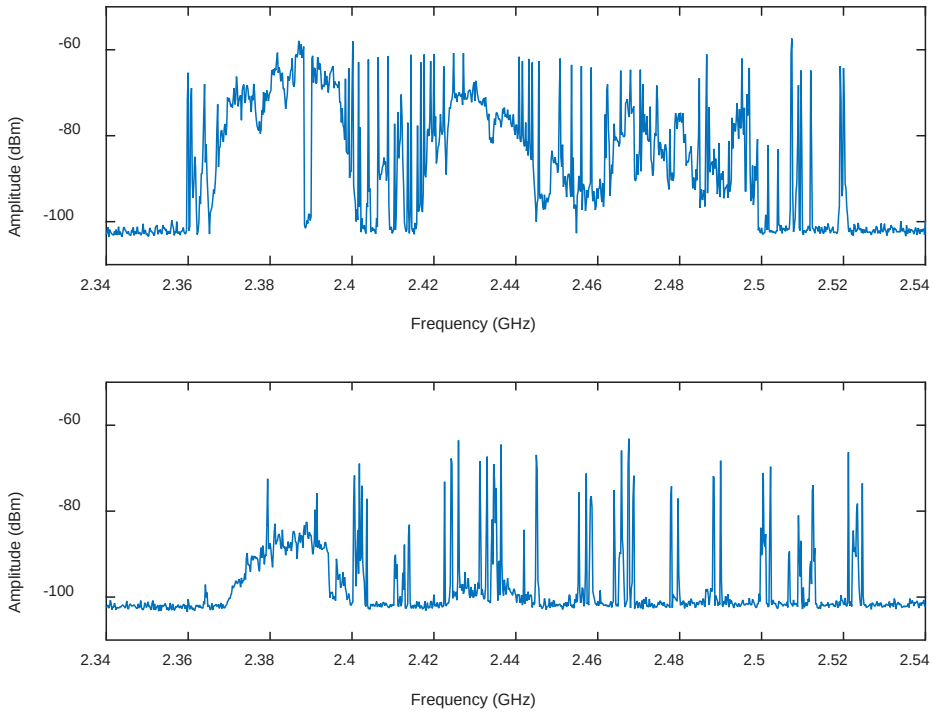

Fig. 21 Indoor Wi-Fi spectrum (top) and outdoor Wi-Fi spectrum (bottom)

Using the same parameters as the first outdoor test, a second outdoor test was performed, this time walking the mobile unit down a slowly sloping hill. Figures 22–25 show the results of this test, corresponding to Figs. 17–20 of the first outdoor test. In general, the results here were similar to the first outdoor test, with a mean error of about 1 m and a small variance. Performance dropped off dramatically past 150 m, whereas in the first test, dramatic error was only observed past 170 m. This can be explained by the more challenging environment posed by the sloping hill.

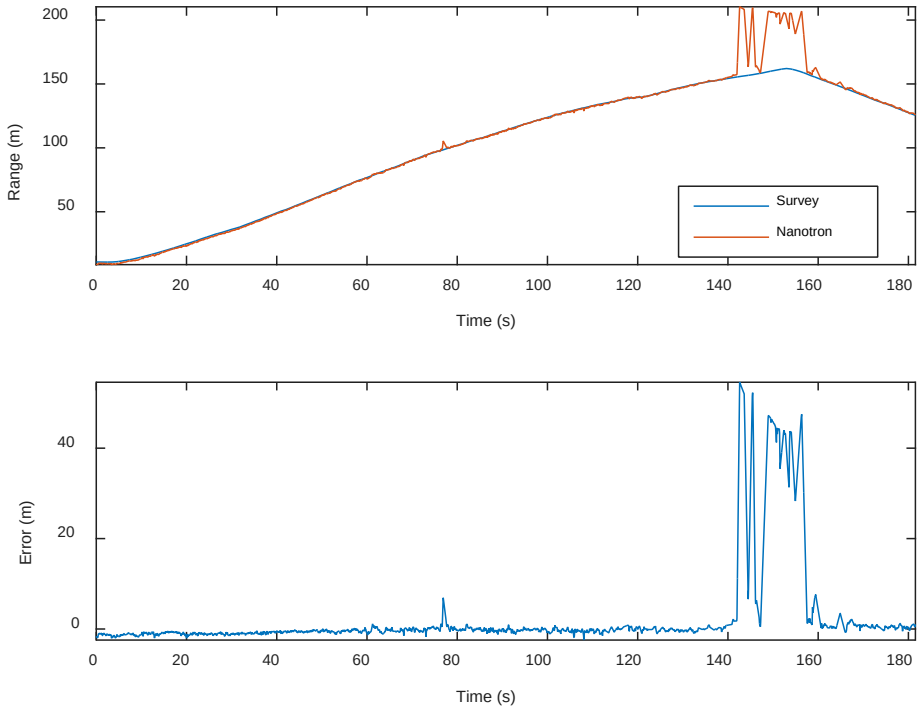

Fig. 22 Comparison of the Nanotron and survey ranges of the second outdoor test (top) with the calculated error (bottom)

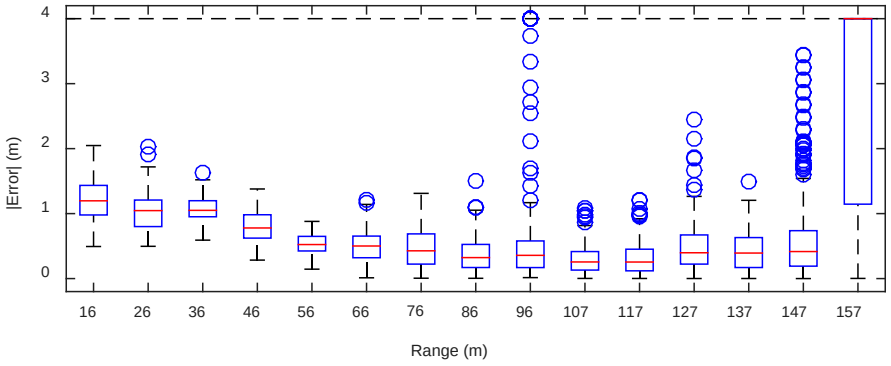

Fig. 23 Boxplot of the absolute value of the error of the second outdoor test

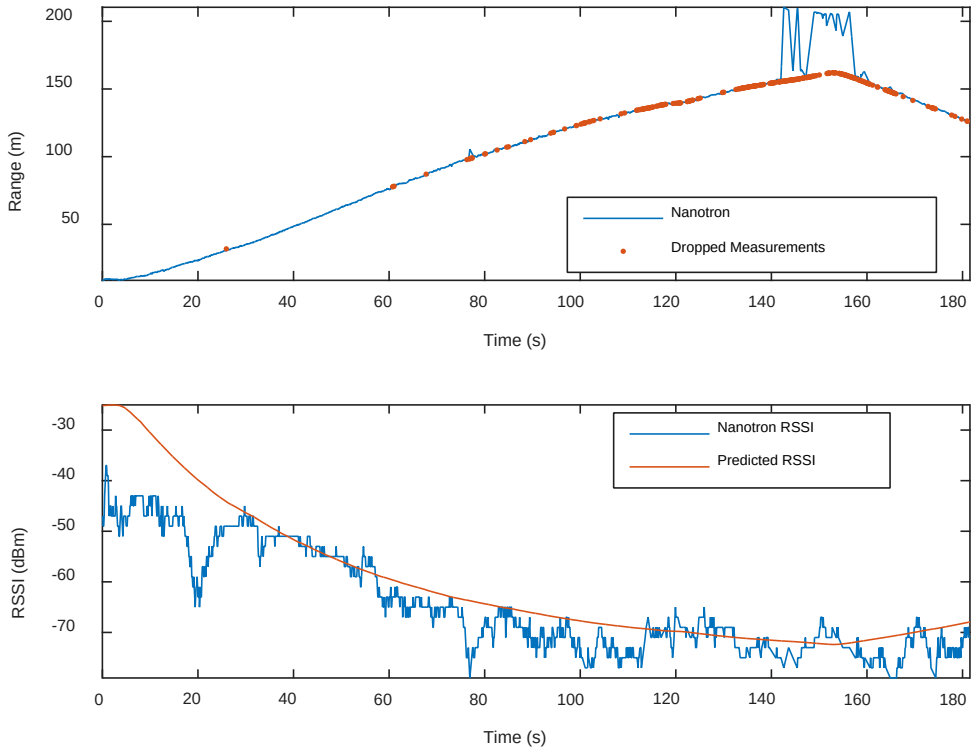

Fig. 24 Dropped Nanotron ranges (top) and RSSI (bottom) of the second outdoor test

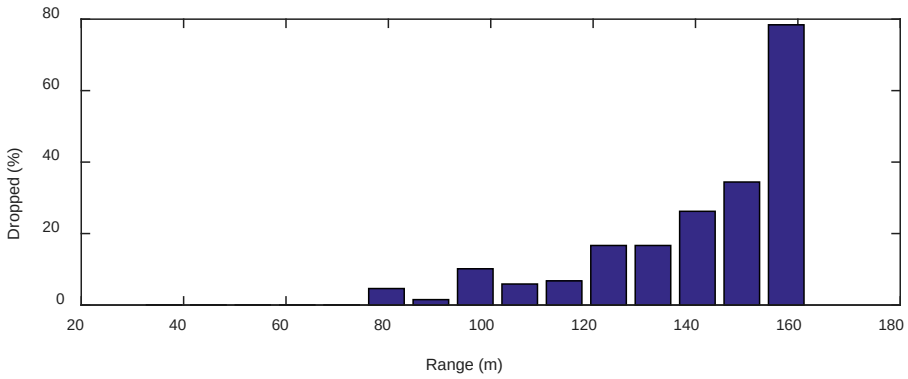

Fig. 25 Percentage of dropped ranges vs. range of the second outdoor test

A third outdoor test was performed, this time with slightly higher dynamics, by running the mobile unit toward and away from the stationary unit. The survey sampling rate was increased to 5 Hz to accommodate the faster motion. Figures 26–29 show the results of this test. Again, the results were similar, with an average range error of about 1 m. One notable difference is the high number of dropped measurements as the mobile unit is moving away from the stationary unit in the top plot of Fig. 28. These correspond to noticeably lower RSSI levels in the

bottom plot, even though they occur at relatively short distances. It appears that the body of the person carrying the mobile unit blocked the line of sight between the 2 Nanotrons, causing a decrease in performance. A large number of measurements were also dropped at the end of the test when the 2 units were very close together. It is likely that this was caused by using the highest TX power level at such a close distance as indicated by the clipped reported RSSI and the high predicted RSSI levels.

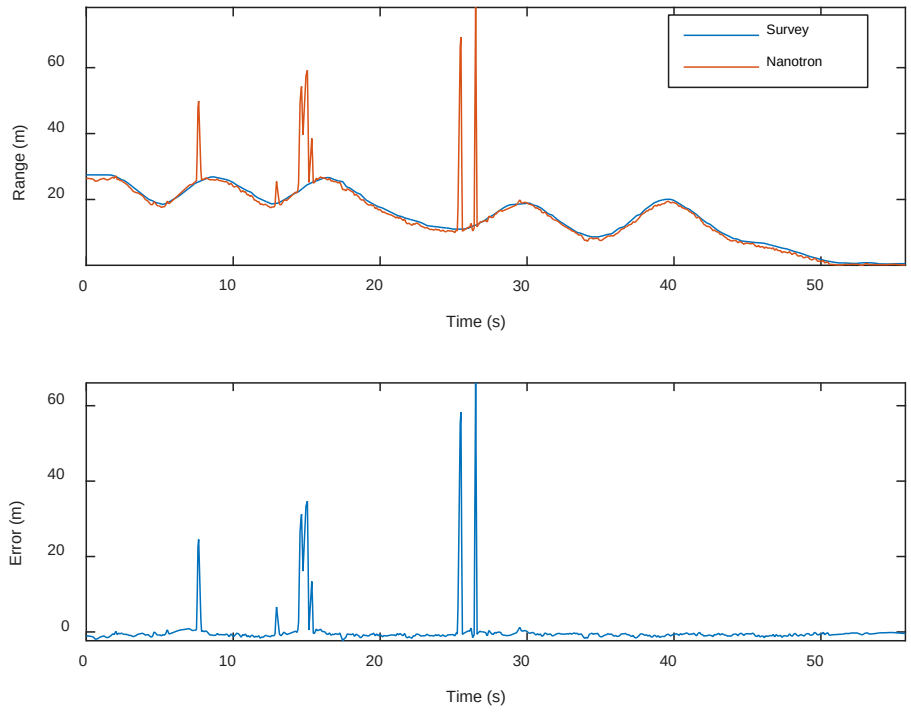

Fig. 26 Comparison of the Nanotron and survey ranges of the third outdoor test (top) with the calculated error (bottom)

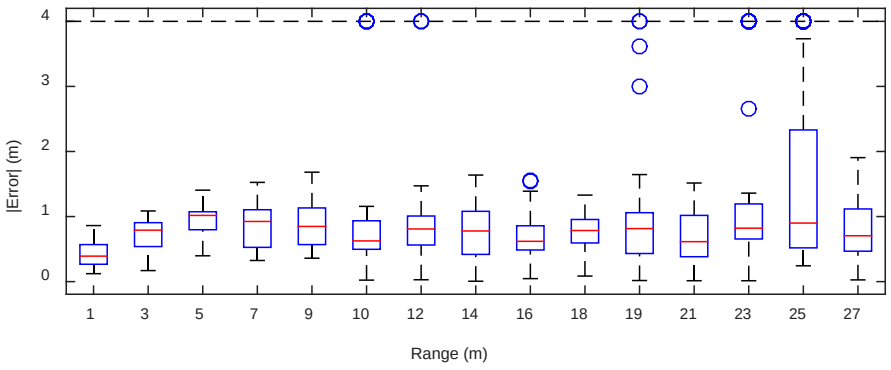

Fig. 27 Boxplot of the absolute value of the error of the third outdoor test

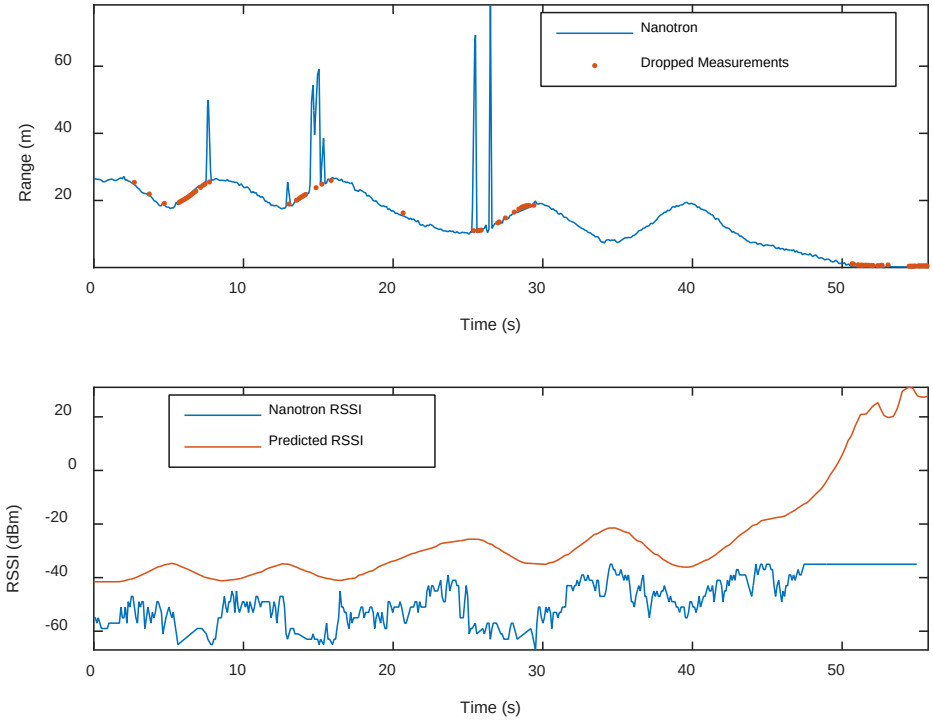

Fig. 28 Dropped Nanotron ranges (top) and RSSI (bottom) of the third outdoor test

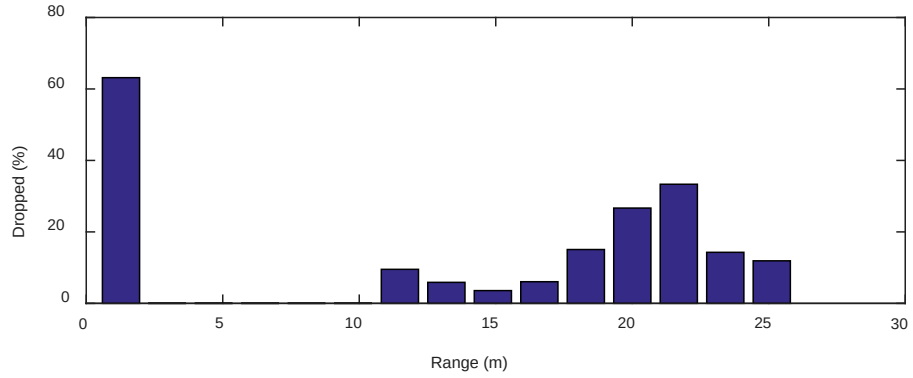

Fig. 29 Percentage of dropped ranges vs. range of the third outdoor test

3.4 Nanotron Full Swarm Localization Testing

After testing ranging between 2 Nanotrons, additional Swarm BEE LE development boards were purchased to evaluate full swarm ranging and localization for 6 agents. The development boards, shown in Fig. 30, are small breakout boards for the Swarm BEE LE module that have connections to only a few essential inputs/outputs (I/Os) as compared with the larger kits used in the initial testing shown in Fig. 7.

Fig. 30 Nanotron Swarm BEE LE development board

For the full swarm localization tests, a TDMA scheme was used that was slightly different from the one proposed in Table 6. Each agent ranges to all of the other agents and broadcasts all the ranging results together in a combined broadcast. The total number of ranging operations is now N^2. As before, the next agent's turn is cued by the previous agent's broadcast or a timeout if the broadcast is missed. The additional range operations increases the time it takes for the entire swarm to complete a cycle of all of the ranging operations, but some time is saved by each agent broadcasting all of its ranges together in one broadcast, giving a total swarm ranging rate of about 3 Hz for 6 agents. The additional ranging operations also make the system more robust to dropped measurements. The Arduino program for this full swarm ranging scheme is listed in Appendix B.

The unit setup for the full swarm localization testing is shown in Fig. 31, with the Nanotron Swarm BEE LE development board, Arduino Mega controller, 11.1-V lithium battery, 5-V regulator, and 2.4-GHz antenna indicated. The Arduino has its own 5-V regulator, but using this regulator reduced the reliability of the UART communications between the Arduino and Nanotron board. The Arduino I/O operates at 5 V, while the Nanotron I/O operates at 3 V, which, although it works reliably most of the time, can result in communication problems. Using an external 5-V regulator produced a slightly lower voltage onboard the Arduino, improving UART performance. The default Nanotron 2.4-GHz antenna was replaced with a Taoglas FXP73 Blue Diamond 2.4-GHz band antenna,[24] which improved ranging reliability. LED indicators were added to the Arduino boards so that their current state of operation could be easily determined.

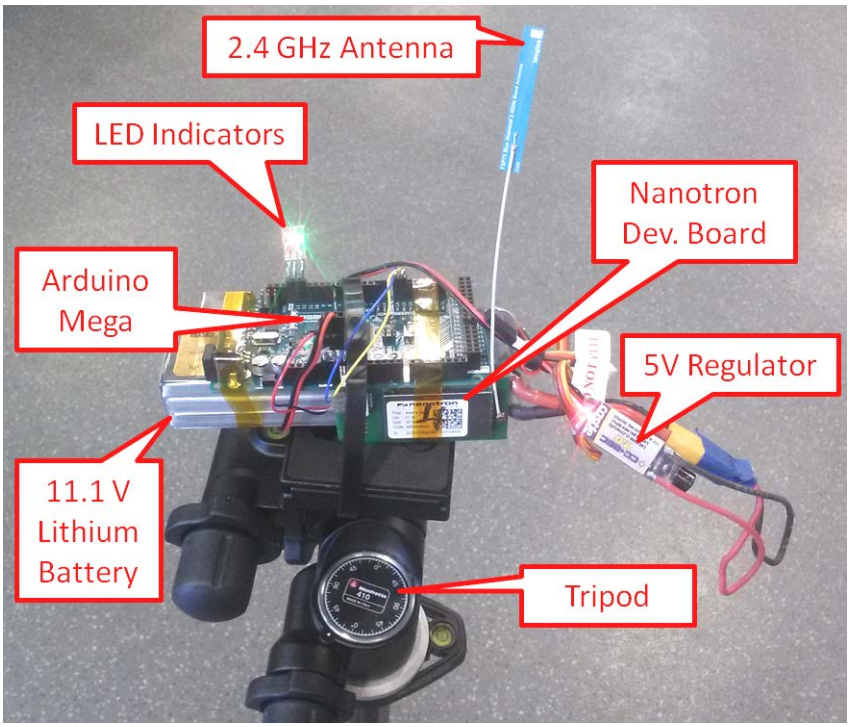

Fig. 31 Unit setup for full swarm localization testing

Figure 32 shows the test area for the full swarm localization experiment with the placement of each agent marked. A2 and A3 were places on tripods, and A5 and A6 were placed on ladders. A3 was placed on a cart with a PC to record the ranging data. A1 was the mobile unit that was hand carried for these swarm localization tests.

Fig. 32 Full swarm localization test area with agent locations marked

3.4.1 Multidimensional Scaling

Classical multidimensional scaling (C-MDS)[25–27] was used to determine the swarm relative localization from the range data in postprocessing. MDS is a class of techniques for projecting high-dimensional data in $\mathbb{R}^N$ feature space onto a low-dimensional space, typically $\mathbb{R}^2$ or $\mathbb{R}^3$.[28] The data are collected from the $\binom{N}{2}$ pairwise proximity measurements between N objects (e.g., nations, candidates, and medications) under some defined metric. The task of MDS is then to find a geometric embedding whose pairwise distances most closely match those in the feature space. The method was originally developed by mathematical psychologists to facilitate data analysis and visualization but has since found application in other disciplines.

C-MDS assumes that the proximity measurements are Euclidean distances and seeks to find a consistent geometric point configuration. The method is noniterative with complexity of approximately $O(N^3)$, where N is the number of agents to be localized. More-accurate iterative localization techniques exist but typically suffer from an order of magnitude or larger increase in complexity. As an example, scaling by majorizing a complicated function[29,30] guarantees monotonic convergence but has complexity $O(N^3 + N^{3/2}t)$, where the additional parameter t is the number of iterations.[31] Due to slow convergence and the existence of local minima, many iterations over multiple initializations are typically needed to find an optimal fit.

For real-time dynamic localization, C-MDS therefore has a marked advantage over iterative localization techniques in terms of raw speed—provided N is not too large.

Unlike iterative methods that easily accommodate arbitrary weighting schemes, C-MDS gives equal weight to all measurements. Because C-MDS requires all pairwise measurements, missing measurements d_{ij} must be somehow estimated. Shang et al.[27] proposed using an all-pairs shortest path algorithm (e.g., Dijkstra[32] or Floyd-Warshall[33]) to complete the distance matrix. Shortest path distance is, however, a maximum bound on the Euclidean point-to-point distance. If the measurement mechanism is itself range-limited, then this threshold can be taken to be the lower bound on missing measurements. d_{ij} can then be better estimated as the average of the sensor range and shortest path distance. Alternatively, if measurement occlusion occurs sporadically within the sensing range of the device, the most recently measured d_{ij} could be used provided the update rate is fast compared with the relative velocities of the agents.

Without so-called "anchor" agents whose location is known with respect to some fixed coordinate frame, the embedding solution is unique only up to translation, rotation, and reflection. The use of average geometric misalignment of true to estimated coordinates as a metric for localization performance is therefore complicated by the additional estimation step of finding an optimal rigid transformation to bring the 2-point sets into alignment. This transformation, which is not strictly part of the relative localization solution, may introduce additional error and lead to misleading results. In general, a more suitable metric must evaluate the fit without reference to any particular transformation. One such metric is the Kruskal stress,[26,28] which directly compares measured distances with those calculated from the estimated embedding. Results from rigidity theory then guarantee that if all pairwise distances are the same, then their respective embeddings must also be identically unique, up to congruence. However, since only one agent moves in these experiments, we shall rely on the remaining $N-1$ stationary agents to define a persistent coordinate frame and transform to this frame to directly evaluate misalignment.

The procedure for C-MDS is as follows.[34] Let $\boldsymbol{D}^{\odot 2} \in \mathbb{R}^{N\times N}$ be the squared Euclidean distance matrix composed of elements d_{ij}^2 representing the squared distance from agent i to agent j. $(.)^{\odot 2}$ denotes the Hadamard (element-wise) exponentiation. $\boldsymbol{D}^{\odot 2}$ is double centered by

$$\boldsymbol{B} = -\frac{1}{2}\boldsymbol{C}\boldsymbol{D}^{\odot 2}\boldsymbol{C}\,, \tag{7}$$

where C is the centering matrix

$$\boldsymbol{C} = \boldsymbol{I} - \frac{1}{N}\mathbb{O}. \tag{8}$$

$\boldsymbol{I} \in \mathbb{R}^{N\times N}$ is the identity matrix, and $\mathbb{O} \in \mathbb{R}^{N\times N}$ is a matrix of all 1's. The agent location matrix $\boldsymbol{X}_3 \in \mathbb{R}^{N\times 3}$ is then the first 3 columns of $\boldsymbol{X}$ given by

$$\boldsymbol{X} = \boldsymbol{E}\boldsymbol{\Lambda}^{\odot 1/2}, \tag{9}$$

where $\boldsymbol{E}$ is a matrix of the N eigenvectors of $\boldsymbol{B}$, and $\boldsymbol{\Lambda}$ is a diagonal matrix of the corresponding N eigenvalues of $\boldsymbol{B}$ in descending order. $\boldsymbol{X}_3$ is the relative localization of the swarm agents; however, successive localization calculations may differ in rotation and translational components. To plot the path of A1 in the same absolute reference frame, the first calculated $\boldsymbol{X}_3$ was designated as a reference, and all other $\boldsymbol{X}_3$ samples were rotated and translated to minimize the root-mean-square error to the reference points using the Kabsch algorithm.[35] Since A1 was moving, and all of the other agents were stationary, all of the agents except for A1 were used to determine the rotation and translation. This subset containing all of the stationary agents at the reference locations is designated $\boldsymbol{S}_1$, while the locations to be rotated and translated are $\boldsymbol{S}_2$. The Kabsch algorithm starts by calculating and subtracting the centroids of $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ giving

$$\boldsymbol{P}_1 = \boldsymbol{S}_1 - \frac{1}{N}\mathbb{O}\boldsymbol{S}_1 \tag{10}$$

and

$$\boldsymbol{P}_2 = \boldsymbol{S}_2 - \frac{1}{N}\mathbb{O}\boldsymbol{S}_2. \tag{11}$$

Next, the cross covariance matrix is calculated as

$$\boldsymbol{A} = \frac{1}{N}\boldsymbol{P}_1^T\boldsymbol{P}_2. \tag{12}$$

Using singular value decomposition, $\boldsymbol{A}$ is represented as

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{W}\boldsymbol{V}^T. \tag{13}$$

The rotation matrix is then

$$\boldsymbol{R} = \boldsymbol{U}\boldsymbol{V}^T \tag{14}$$

and the translation is

$$\boldsymbol{t} = \frac{1}{N}\boldsymbol{S}_1^T\mathbf{1} - \frac{1}{N}\boldsymbol{S}_2^T\mathbf{1}\boldsymbol{R}, \tag{15}$$

where $\mathbf{1} \in \mathbb{R}^{N\times 1}$ is a vector of all 1's.

3.4.2 Results

Figure 33 shows the results of the first swarm localization test with the mobile agent moving in a clockwise circle. The stationary agent location samples are marked with asterisks, while the path of the mobile agent is shown with a line. The start and end positions of the mobile agent are marked with a green and a red "X", respectively. Figure 34 shows a 3-D view of Fig. 34. The range data were preprocessed to replace d_{ij} and d_{ji} with their average value. Any dropped measurement was replaced by its corresponding measurement pair or nearest successful ranging operation in a previous ranging cycle. C-MDS was then employed to find the localizations, and the Kabsch algorithm was used to align all of the location samples to the first localization. The path of A1 was smoothed with a 50-tap moving average filter. Figure 35 shows the results of a second localization test with the mobile agent moving counterclockwise, and Fig. 36 shows the results for a criss-cross test, where A1 was carried is a criss-cross pattern between the stationary agents. Figure 37 shows the standard deviations of the calculated positions of the stationary agents. All of these results show a low variance with standard deviations on the order of 1 m.

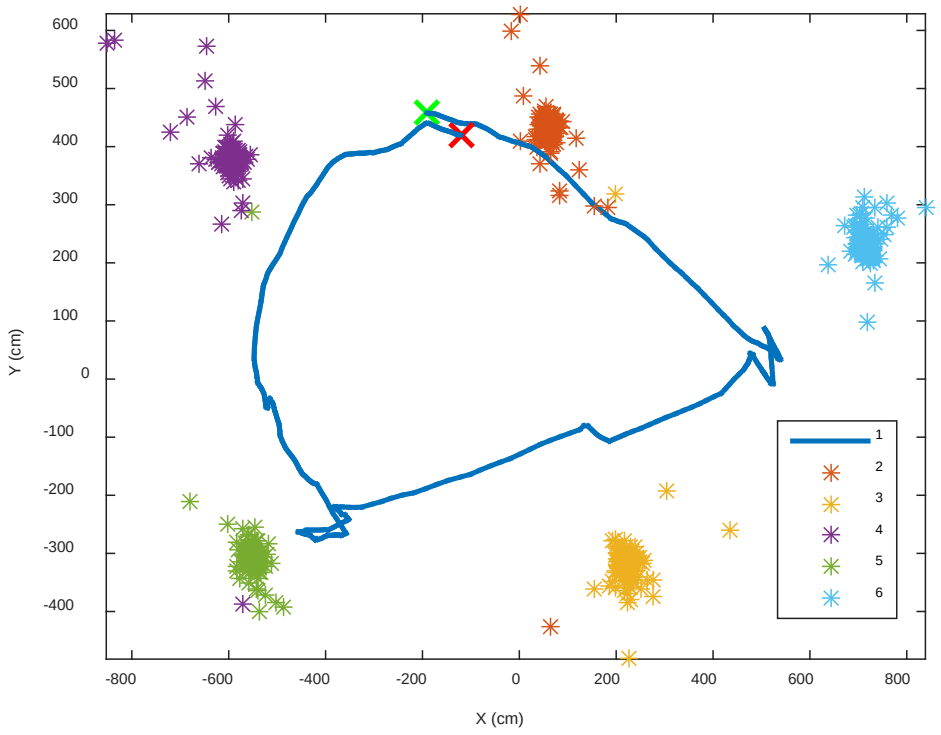

Fig. 33 Full swarm localization clockwise test results, 2-D view

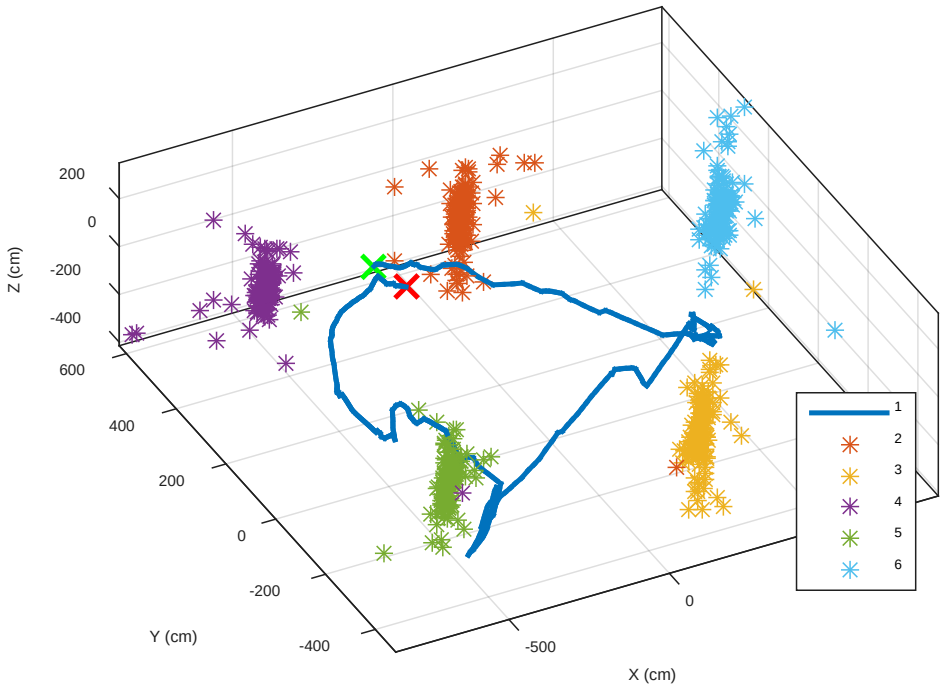

Fig. 34 Full swarm localization clockwise test results, 3-D view

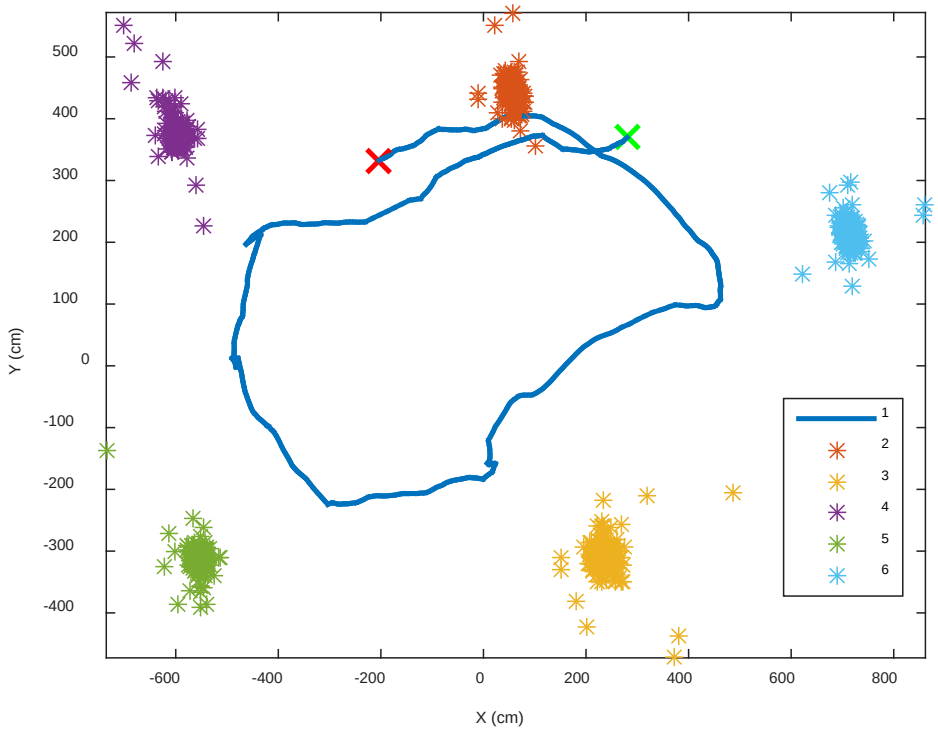

Fig. 35 Full swarm localization counterclockwise test results

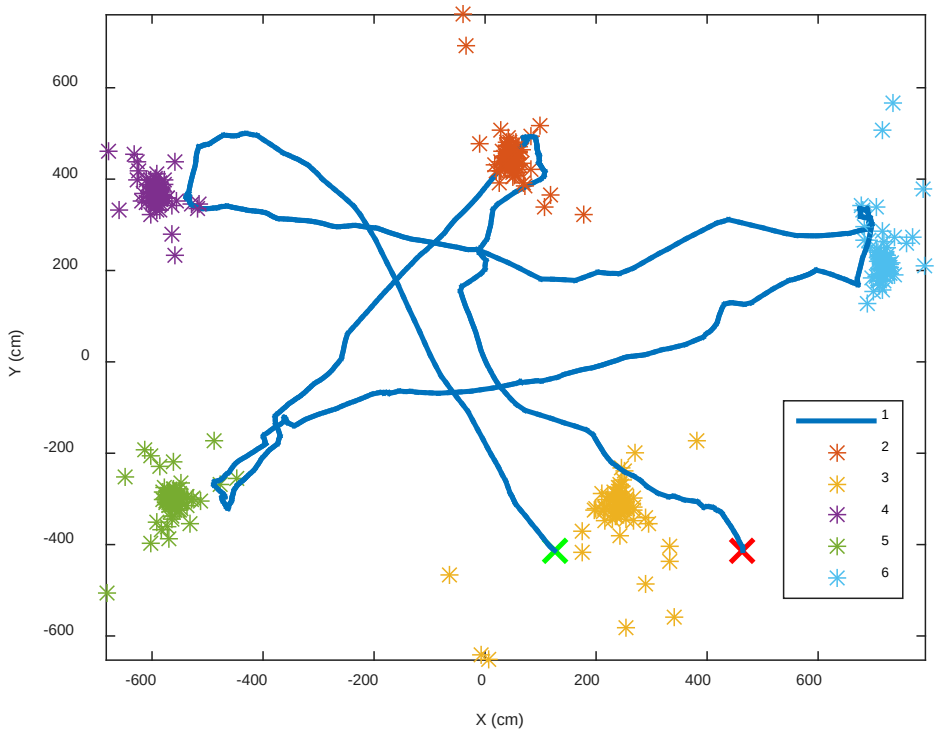

Fig. 36 Full swarm localization criss-cross test results

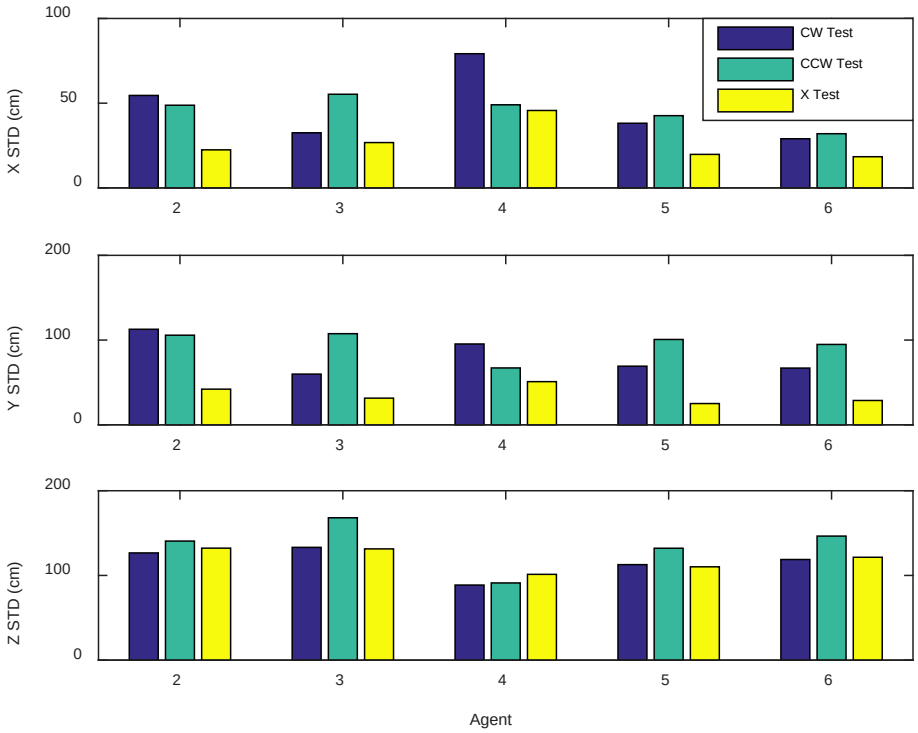

Fig. 37 Standard deviations of X, Y, and Z components of stationary agent locations for the 3 tests

4. Conclusion

Two RF TWR products have been evaluated for swarm localization in GPS-denied environments. The Atmel product was rejected due to its slow sampling rate and poor performance in long-range outdoor testing. The Nanotron transceiver showed promise as a candidate for swarm localization. A dual Nanotron test setup was developed using Arduino controllers and a TDMA networking scheme. Data in both a controlled laboratory environment and an outdoor setting were presented and analyzed, showing acceptable range, accuracy, and measurement rates. In addition, an experiment was successfully carried out for the swarm localization of 6 agents.

Future research should include swarm localization experiments where all of the agents are mobile. This will require the testing infrastructure to monitor the position of multiple agents at the same time. Testing should be conducted in a Wi-Fi-free environment in order to characterize the effect of Wi-Fi on Nanotron performance. Performance should also be evaluated in high-dynamic environments by mounting devices on remote-controlled vehicles. In addition to the current configuration, other hardware platforms can be investigated. The use of external RF amplifiers can extend range, and dual UWB and NB solutions can combine the accuracy of

UWB at short ranges with the long range capabilities of NB. Building upon ARL's experience with software-defined radio,[36,37] custom RF ranging development can also be investigated. There are many avenues to explore, but the initial research in this report has shown the feasibility of RF ranging for swarm localization.

5. References

1. Kurazume R, Nagata S, Hirose S. Cooperative positioning with multiple robots. Proceedings of IEEE International Conference on Robotics and Automation; 1994 May 8–13; San Diego, CA.

2. Bachrach A, Prentice S, He R, Roy N. RANGE – robust autonomous navigation in GPS-denied environments. Journal of Field Robotics. 2011;1:28.

3. Parker R, Valaee S. Vehicular node localization using received-signal-strength indicator. IEEE Transactions on Vehicular Technology. 2007;56.

4. Ryan A, Zennaro M, Howell A, Sengupta R, Hedrick JK. An overview of emerging results in cooperative UAV control. Proceedings of the 43rd IEEE Conference on Decision and Control; 2004. doi: 10.1109/CDC.2004 .1428700.

5. Fresconi F, Fermen-Coker M. Delivery of modular lethality via a parent-child concept. Proceedings of the AIAA Atmospheric Flight Mechanics Conference; 2015 Feb. p. 2708.

6. Iliev N, Paprotny I. Review and comparison of spatial localization methods for low-power wireless sensor networks. IEEE Sensors Journal. 2015;15(10):5971–5987.

7. Medina C; Segura JC, de la Torre A. A synchronous TDMA ultrasonic TOF measurement system for low-power wireless sensor networks. IEEE Trans Instrum Meas. 2013;62(3):599–611.

8. Piatti D. Time-of-flight cameras: tests, calibration and multi-frame registration for automatic 3D object reconstruction [PhD dissertation]. [Turin, Italy]: Doctoral School, Politecnico Torino; 2010.

9. Ross B. A practical stereo vision system. CVPR'93. Proceedings of the 1993 IEEE Computer Society Conference on computer vision and pattern recognition; 1993 June [accessed 2017 Sep 5]. http://ieeexplore/ieee.org /xpl/mostRecentIssue.jsp ?punumber=916.

10. Saska M, Vonásek V, Krajník T, Přeučil L. Coordination and navigation of heterogeneous UAVs-UGVs teams localized by a hawk-eye approach. Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2012. doi: 10.1109/IROS.2012.6385517.

11. Don M. The feasibility of radio direction finding for swarm localization. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2017 Aug. Report No.: ARL-TR-8114.

12. Liu J, Wang Q, Wan J, Xiong J. Towards real-time indoor localization in wireless sensor networks. Proceedings of the IEEE 12th International Conference for Computer and Information Technology; 2012 Oct. p. 877–884.

13. Li X. RSS-based location estimation with unknown pathloss model. IEEE Trans Wireless Commun. 2006;5(12):3626–3633.

14. Lanzisera S, Pister K. RF ranging for location awareness [PhD dissertation]. [Berkeley (CA)]: University of California Berkeley; 2009.

15. Fontana RJ, Richley E, Barney J. Commercialization of an ultra wideband precision asset location system. Proceedings of the IEEE Conference for Ultra Wideband System Technology; 2003 Nov. p. 369–373.

16. AT86RF233 datasheet, Atmel Corporation [accessed 2017 Sep 5]. http://www.atmel.com/images/Atmel-8351-MCU_Wireless-AT86RF233 _Datasheet.pdf.

17. Nanotron Swarm BEE LE datasheet [accessed 2017 Sep 5]. http://top -electronics.com/userfiles/swarmbeeLEPBV1.3May2016.pdf.

18. Wing MG, Eklund A, Kellogg LD. Consumer-grade global positioning system (GPS) accuracy and reliability. Journal of Forestry. 2005;103.4:169–173.

19. Atmel AT03911: REB233CBB module – user manual [accessed 2017 Sep 5]. http://www.atmel.com/Images/Atmel-42156-WIRELESS-AT03911 -REB233CBB-Module-User-Manual_Application-Note.pdf.

20. Critical golf: unbiased equipment reviews [accessed 2017 Aug 14]. http://www.criticalgolf.com/laser-rangefinder-accuracy.

21. Arduino mega [accessed 2017 Sep 5]. https//www.arduino.cc/en/Main /arduinoBoardMega.

22. Goldsmith A. Wireless communications. Cambridge (UK): Cambridge University Press; 2005. p. 31

23. Leica Viva TS16 - Robotic Total Station [accessed 2017 Oct 19]. http://leica -geosystems.com/en-us/products/total-stations/robotic-total-stations/leica-viva-ts16.

24. Blue Diamond FXP73 2.4GHz flex PCB antenna [accessed 2017 Sep 5]. http://www.taoglas.com/wp-content/uploads/2015/08/FXP73.09.0100A.pdf.

25. Cox TF, Cox MAA. Multidimensional scaling. 2nd ed. Boca Raton (FL): Chapman & Hall/CRC; 2001.

26. Shang Y, Ruml W. Improved MDS-based localization. IEEE INFOCOM. 2004;4:2640–2651.

27. Shang Y, Ruml W, Zhang Y, Fromherz MPJ. Localization from mere connectivity. Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing; 2003. p. 201–212.

28. Borg I, Groenen PJF. Modern multidimensional scaling: theory and applications. 2nd ed. New York (NY): Springer; 2005.

29. de Leeuw J, Heiser WJ. Convergence of correction matrix algorithms for multidimensional scaling. Ann Arbor (MI): Mathesis Press; 1977.

30. de Leeuw J, Mair P. Multidimensional scaling using majorization: SMACOF in R Jour of Stat Soft. 2009;31(3).

31. Asano T, Bose P, Carmi P, Maheshwari A, Shu C, Smid M, Wuhrer S. A linear-space algorithm for distance preserving graph embedding. Computational Geometry. 2009;(42):289–304.

32. Dijkstra, E. W. A note on two problems in connexion with graphs. Numerische Mathematik. 1959;1:269–271. doi:10.1007/BF01386390.

33. Floyd, Robert W. Algorithm 97: shortest path. Communications of the ACM. 1962;5(6):345. doi:10.1145/367766.368168.

34. Wickelmaier F. An introduction to MDS. Aalborg (Denmark): Sound Quality Research Unit, Aalborg University; 2003.

35. Kabsch W. A solution for the best rotation to relate two sets of vectors. Acta Crystallographica Section A. 1976;32:922–923.

36. Don M. A low-cost software-defined telemetry receiver. Proceedings of the 51st International Telemetering Conference; 2015 Oct 26–29; Las Vegas, NV.

37. Don M. Advances in a low-cost software-defined telemetry system. Presented at the 53rd International Telemetering Conference Proceedings; 2017 Oct; Las Vegas, NV.

Appendix A. Dual Nanotron Ranging Arduino Program

This appendix appears in its original form, without editorial change.

```
String ID;//saves local ID from node
String IDs[] = {"0000652F0158", "000003BA5797"};// these two IDs
are the wones ranging to eachother
int myIndex = -1;//index = 0 or 1 for local ID
int otherIndex = -1;
boolean Ranging = false;//doing ranging methods
double dist1, dist2;//saved distances between the two
long cycleTime;


void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  Serial3.begin(115200);

  Serial3.write("SFAC\r\n");//restore defaults
  delay(100);
  Serial.println(getMsg());
  Serial3.write("SUAS 500000\r\n");//UART speed
  delay(100);
  Serial.println(getMsg());
  Serial3.flush();
  delay(2);
  Serial3.end();
  Serial3.begin(500000);
  delay(2);
  Serial3.write("SFEC 0\r\n");    //1 or 0 error correction
  delay(100);
  Serial.println(getMsg());
  Serial3.write("SDAM 1\r\n");          //1 or 2 data rate
  delay(100);
  Serial.println(getMsg());
  Serial3.write("EBID 0\r\n");//turns off random node ID blink
  delay(100);
```

```
  Serial.println(getMsg());
  Serial3.write("SROB 0\r\n");//turns of random ranging blink
  delay(100);
  Serial.println(getMsg());
  Serial3.write("STXP 20\r\n");//set TX power
  delay(100);
  Serial.println(getMsg());
  Serial3.write("GNID\r\n");//get ID
  delay(100);
  ID = getMsg();

  //get just the ID string
  ID.replace("=", "");
  ID.replace("\n", "");
  ID.replace("\r", "");
  Serial.println(ID);

  // checks to see if the ID is a match
  if (ID.equals(IDs[0]))
  {
    //saves ID and tells it that it should start the ranging
first
    Ranging = true;
    myIndex = 0;
    otherIndex = 1;
  }
  else
    {
      myIndex = 1;
      otherIndex = 0;
    }
    //user has to type in a at the same time with both programs
so that they start ruffly at the same time
    while (!getInput().equals("a\n"))
```

```
      delay(100);
  Serial.println("GO");
  cycleTime = millis();
}


void loop() {
  if (Ranging)
  {
    Serial3.write(convertString("RATO 0 " + IDs[otherIndex] +
"\r\n"));//range to other node
    while (!Serial3.available());//waits for message
    dist1 = getRATO();//takes range
    Ranging = false;//start recieving
  }
  else if (Serial3.available())//message available
  {
    dist2 = getRange();//takes range broadcast
    Ranging = true;
    cycleTime = millis();//update time
  }
  else if ((millis() - cycleTime) / 2 > 20)//if the time is
taking longer than 20 ms to recieve then range again
  {
    Serial.println("Timeout");
    cycleTime = millis();//update time
    Ranging = true;
  }
}


String getInput()
{
  String send = "";
  while (Serial.available())
    send = send + (char)Serial.read();
```

```
  return send;
}
char* convertString(String pass)
{
  char *p = const_cast<char*>(pass.c_str());
  return p;
}

String getMsg()
{
  String send = "";
  while (Serial3.available())
    send = send + (char)Serial3.read();
  return send;
}

double getRange()
{
  String send = "";
  while (Serial3.available())
    send = send + (char)Serial3.read();
  send = send.substring(33, 39);
  return send.toInt() / 100.00;
}

double getRATO()
{
  String send = "";
  while (Serial3.available())
    send = send + (char)Serial3.read();
  send = send.substring(3, 9);
  return send.toInt() / 100.00;
}
```

INTENTIONALLY LEFT BLANK.

Appendix B. Full Swarm Nanotron Ranging Arduino Program

This appendix appears in its original form, without editorial change.

```
int green = 12;
int red = 13;


const int swarmSize = 6;//set the amount of nodes you will have for
your swarm
const int period = 330;//set the wait period if msg is missed


int myID = 0;//ID of local node
String ID[25];//makes a list of all the IDs in the swarm
long startTime;//time of the program starting
boolean runIf = true;//condition to start local broadcast
const int timeDelay = 6;//set time delay


void setup() {
  // put your setup code here, to run once:
  //LEDs
  pinMode(green, OUTPUT);
  pinMode(red, OUTPUT);
  digitalWrite(green, LOW);
  digitalWrite(red, HIGH);
  Serial.begin(115200);
  Serial3.begin(115200);
  //loop condition of the Arduino calling the local ID from the
Nanotron
  while (myID == 0)
  {
    delay(100);
    Serial3.write("GNID\n");
    while (!Serial3.available());
    String myIDmsg = getMsg();
    myIDmsg.replace("=", "");
    myIDmsg.replace("\n", "");
    myIDmsg.replace("\r", "");
    myID = myIDmsg.toInt();
```

```
  }


  //initializing the ID array to have all the IDs in the swarm
  int m = 0;
  for (int i = 1; i <= swarmSize; i++)
  {
    if (i != myID)
    {
      if (i < 10)
        ID[m] = "00000000000" + (String)i;
      else
        ID[m] = "0000000000" + (String)i;
      m++;
    }
  }


  //LED state
  digitalWrite(green, HIGH);


  //condition to have all the nodes start void loop() at the same
time
  if (myID == 1)//parent (ID one) will broadcast this message to be
received by the childeren and void loop() will begin
  {
    Serial3.write("BDAT 0 3 010203\n");
    startTime = millis();
  }
  else//childeren wait for this message to be received and when it
does void loop() will begin
  {
    String start = "";
    while (start != "=03 000000000001 010203\r\n")//received msg of
BDAT from node 1
    {
      while (!Serial3.available())//waiting for message
```

```
      {
        //LED state
        digitalWrite(green, HIGH);
        digitalWrite(red, LOW);
        digitalWrite(green, LOW);
        digitalWrite(red, HIGH);
      }
      getMsg();//gets *DNO
      Serial3.write("GDAT\n");//get message
      while (!Serial3.available());
      start = getMsg();//receives message
      startTime = millis() - timeDelay;//intializes the start
offset for loop()
    }
  }


  //LED
  digitalWrite(green, HIGH);
  digitalWrite(red, LOW);
}
void loop() {
  int swarm_cycle_time = Time() % (period * swarmSize); //current
time in the cycle that is relative to all the nodes
  int local_start_time = (myID - 1) * period; //the time that the
node should start broadcasting
  int local_end_time = myID * period; //the time the node should
finish broadcasting


  //if time is within broadcast range and runIF is true then range
and broadcast msg
  //broadcast ranges to all the nodes and then sends the results.
  //runIf will become false
  if (swarm_cycle_time > local_start_time && swarm_cycle_time <
local_end_time && runIf)
    broadcast();
```

```
  //not in time range to broadcast but msg are coming in
  //receive receives the broadcast from other nodes and updates the swarms local time
  //based on which node just broadcast so it doesn't have to wait the full period time everytime
  else if (Serial3.available())
    receive();

  //this is set so that the first condition only call broadcast() one time within its time range
  //without this, the Broadcast method would keep running for the period time
  else if ((swarm_cycle_time < local_start_time || swarm_cycle_time > local_end_time) && !runIf)
    runIf = true;
}
//gets msg from nanotron
String getMsg()
{
  String send = "";
  while (Serial3.available())
    send = send + (char)Serial3.read();
  return send;
}
//allows to send strings as a command
char* convertString(String pass)
{
  char *p = const_cast<char*>(pass.c_str());
  return p;
}
//time difference between the arduino time and the time saved from either starting the program or received messages
long Time()
{
  return (millis() - startTime);
}
```

```
//ranges and sends all the messages to the other nodes
void broadcast()
{
  //LED
  digitalWrite(red, HIGH);


  String printDist = "";//string printed out for the Cpp code to
read
  String Broadcast = "";//string built to send as a command to the
Nanotron
  for (int i = 0; i < swarmSize - 1; i++)
  {
    Serial3.write(convertString("RATO 0 " + ID[i] + "\n"));//to all
the IDs
    String msg;
    while (!Serial3.available());
    msg = getMsg();
    printDist = printDist + (ID[i] + "\t" + msg + "!");
    Broadcast = Broadcast + msg.substring(3, 9);
  }
  Broadcast = "BDAT 0 " + String(Broadcast.length() / 2, HEX) + "
" + Broadcast + "\n";//command for Nanotron to broadcast
  Serial3.write(convertString(Broadcast));
  digitalWrite(red, LOW);//led
  int timer = Time();
  Serial.println(printDist                                          +
"/received_at_"+(String)millis()+"|");
  while (!Serial3.available());//broadcast cmd returns a msg that
it sent
  getMsg();
  runIf = false;
}


void receive()
{
  String msg = getMsg();
```

```
  if (msg.substring(0, 5) == "*DNO:")//received broadcast indicator
  {
    String IDmsg = msg.substring(5, 17);//extracts ID
    //resets the start time to skip to the end of that nodes time
range so
    //all the nodes don't have to wait a full period to do the next
range
    int addTime;
    addTime = IDmsg.toInt() * period;
    startTime = millis() - timeDelay * 2 - addTime;

    //gets the data and prints it out
    Serial3.write("GDAT\n");
    while (!Serial3.available());
    msg = getMsg();
    Serial.print(msg + "/");

    Serial.println("recieving_at_" + (String)millis() + "|");
  }
}
```

INTENTIONALLY LEFT BLANK.

List of Symbols, Abbreviations, and Acronyms

2-D	2-dimensional
3-D	3-dimensional
Ai	agent i
AOA	angle of arrival
ARL	US Army Research Laboratory
C-MDS	classical multidimensional scaling
CSS	chirp spread spectrum
FOV	field of view
FSPL	Free-Space Path Loss
GPS	Global Positioning System
ID	identification
IDE	integrated development environment
IO	input/output
ISM	industrial, scientific, and medical
LE	low energy
MAC	medium access control
NB	narrowband
PC	personal computer
PMU	phase difference measurement unit
RF	radio frequency
RSSI	received signal strength indicator
RX	receiver
SDAM	Set Data Mode
SFEC	Switches Forward Error Correction
TDMA	time-division multiple access

TDOA	time-difference-of-arrival
TOA	time of arrival
TWR	2-way ranging
TX	transmitter
UART	universal asynchronous receiver-transmitter
UWB	ultra-wideband

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
2 (PDF)	DIR ARL RDRL DCM IMAL HRA RECORDS MGMT RDRL DCL TECH LIB
1 (PDF)	GOVT PRINTG OFC A MALHOTRA
25 (PDF)	DIR ARL RDRL WML F B ALLIK B ACKER T BROWN S BUGGS E BUKOWSKI J COLLINS J CONDON B DAVIS M DON D EVERSON D GRZYBOWSKI R HALL J HALLAMEYER M HAMAOUI T HARKINS M ILG B KLINE J MALEY C MILLER P MULLER B NELSON D PETRICK K PUGH N SCHOMER B TOPPER

INTENTIONALLY LEFT BLANK.